

گزارش کار پروژه کارشناسی

یک پیام رسان موبایل با اندروید

استاد راهنما : آقای دکتر ناصر فرج زاده

زهرا هاشم زاده : 941832244

زمستان 98

**فهرست**

**مقدمه ...................................................................................................................................................................................4**

**سیستم عامل اندروید ------------------------------------------------------------------5**

**زبان های برنامه نویسی اندروید ......................................................................................................................................6**

**زبان کاتلین ........................................................................................................................................................................16**

**آماده سازی محیط و ابزار ها .......................................................................................................................................23**

**توضیح برنامه .....................................................................................................................................................................24**

# مقدمه

با توجه به گسترش استفاده از پیام رسان های مختلف هدف من طراحی که پیامرسان است که بتوانم از آن استفاده کنم

من برای نوشتن پروژه ما از زبان کاتلین و محیط برنامه نویسی اندروید استدیو استفاده می کنم .

اندروید استدیو یک محیط متن باز است که توسط گوگل طراحی شده متاسفانه دانلود این محیط برای ما توسط خود گوگل تحیریم هستیم برای دانلود باید از نرم افزار های تغیر IP استفاده کنیم یا می توانیم از سایت شکن برای تغیر IP و دسترسی به سایت های تحریم شده استفاده کنیم .

و از سیستم firebase گوگل نیز می توانیم برای دیتابیس و ارسال نوتفیکشن استفاده کنیم . وباز این هم سیستم مانند بسیاری از خدمات گوگل برای IP های ایران تحریم است و باید از ابزار دور زدن تحریم ها استفاده کنیم .

# سیستم عامل اندروید

تصور دنیای بدون اندروید کار چندان ساده‌ای نیست. تقریبا ۱۰ سال از عرضه‌ی اندروید می‌گذرد و حالا به لطف بیش از ۲ میلیارد کاربر، اندروید به پرطرفدارترین سیستم‌عامل جهان بدل شده است. اما به‌رغم این حضور همه‌جانبه، پرسش «اندروید چیست؟» پاسخ ساده‌ای ندارد. هرچند سیستم‌عامل iOS در گجت‌های مختلف ظاهر یکسانی دارد و فقط هم گجت‌های ساخت اپل از آن بهره می‌برند، اما سیستم عامل اندروید در انواع و اقسام گوشی‌های شرکت‌های مختلف حضور دارد و این اندرویدها عمدتا شباهت چندانی به یکدیگر ندارند. دلیل این عدم شباهت‌ها چیست؟

سیستم عامل اندروید با وجود اینکه یک سیستم‌عامل همه‌گیر محسوب می‌شود، اما بسیاری از مفاهیم آن، به خصوص برای کاربران معمولی، می‌تواند گیج‌کننده باشد. به همین خاطر، در ادامه به تعدادی از سوالات رایج پیرامون این سیستم‌عامل، پاسخ می‌دهیم.

#### **سیستم عامل اندروید چیست؟**

#### **اندروید خالص چیست؟**



احتمالا بارها عبارت «اندروید خالص» به گوش‌تان خورده است. اما اندروید خالص چه نوع اندرویدی است؟ همانطور که از اسم آن پیداست، عبارت اندروید خالص به خالص‌ترین نسخه‌ی اندروید اطلاق می‌شود. این نسخه از اندروید چندان سنگین نیست و برخلاف دیگر نسخه‌های اندروید، اندروید خالص از اپلیکیشن‌های پیش‌فرض زیادی بهره نمی‌برد و فقط تعدادی از اپلیکیشن‌های گوگل در آن تعبیه شده است. اما این سادگی تأثیر منفی بر عملکرد این نسخه نداشته و حتی بر اساس نظر بسیاری از کارشناسان، اندروید خالص یکی از بهترین سیستم‌عامل‌های جهان محسوب می‌شود. از جمله گجت‌های مجهز به این اندروید هم می‌توانیم به گوشی‌های پیکسل گوگل اشاره کنیم.

در زمینه‌ی انتخاب نسخه‌های مختلف اندروید، سلیقه و اولویت‌های شخصی نقش مهمی را ایفا می‌کنند؛ اما شایان ذکر است که اندروید خالص معمولا به‌عنوان سریع‌ترین نسخه‌ی اندروید در نظر گرفته می‌شود. این ویژگی نسخه‌ی خالص اندروید، به سبکی آن برمی‌گردد که باعث می‌شود این سیستم‌عامل با بهره‌گیری از منابع بیشتر، وظایف موردنظر خود را با سرعت بیشتری انجام دهد.

#### **اندروید اوریو چیست؟**



یکی از سوال‌هایی که در ذهن کاربران شکل می‌گیرد این است که اندروید نوقا و اوریو چه تفاوت‌هایی با یکدیگر دارند؟ به‌طور خلاصه و مفید باید بگوییم که گوگل هرساله نسخه‌ی جدیدی از اندروید را عرضه می‌کند. هر نسخه‌ی جدید اندروید دچار تغییراتی می‌شود و معمولا ویژگی‌های جدیدی را به ارمغان می‌آورد. از آنجایی که بیشتر گجت‌های اندرویدی امکان دریافت نسخه‌های جدید را ندارند، به همین خاطر همواره نسخه‌های مختلفی از اندروید مورد استفاده قرار می‌گیرند.

اندروید ۸ اوریو جدیدترین نسخه‌ی اندروید محسوب می‌شود و انتظار داریم که در کنفرانس I/O گوگل در اردیبهشت ۱۳۹۷، شاهد معرفی اندروید ۹ باشیم. از سال ۲۰۰۹، نسخه‌های اندروید با نام شیرینی‌ها و به ترتیب حروف الفبا نام‌گذاری شده‌اند. از آنجایی که نام اندروید ۸ «اوریو» (Oreo) بوده، پس قاعدتا نام اندروید ۹ باید با حرف P آغاز شود.

#### **اندروید گو یا اندروید وان چیست؟**



اگر اهل استفاده از گوشی‌های میان‌رده و پایین‌رده باشید، احتمالا گوش‌تان به عبارت‌های اندروید «گو» (Go) و اندروید «وان» (One) خورده است. اندروید گو یک نسخه از اندروید خالص است که برای سخت‌افزارهای ضعیف تنظیم شده است. مدت‌زمان زیادی از عرضه‌ی اندروید گو نمی‌گذرد و تمرکز این سیستم‌عامل عمدتا معطوف به گجت‌های پایین‌رده در بازارهایی مانند هند و برزیل است. از سوی دیگر، اندروید وان سابقه‌ی بیشتری از اندروید گو دارد و می‌توان آن را نسخه‌ی سبک‌تر اندروید خالص به‌حساب آورد. با وجود عرضه‌ی اندروید گو، اندروید وان هنوز از رده خارج نشده و گوشی‌هایی مانند اچ‌تی‌سی U11 Life و Moto X4 از این نسخه‌ی اندروید بهره می‌برند.

#### **پوسته‌ها و رابط‌های کاربری سازندگان**



اندروید سیستم‌عامل انعطاف‌پذیری محسوب می‌شود و بیشتر سازندگان گوشی‌ها با بهره‌گیری از این مشخصه، تغییراتی را در رابط کاربری اندروید گوشی‌هایشان اعمال می‌کنند. هرکدام از شرکت‌ها در این زمینه از رویکردهای متفاوتی بهره می‌برند. از بین این رابط کاربری‌ها می‌توانیم به رابط کاربری Sense UI از شرکت اچ‌تی‌سی و رابط کاربری EMUI از شرکت هوآوی اشاره کنیم. در رابط کاربری Sense UI شاهد تغییر انقلابی‌ای نیستیم و فقط چند عنصر به اندروید خالص اضافه شده است. اما از آن‌طرف هوآوی در رابط کاربری EMUI تغییرات گسترده‌ای را اعمال کرده که ظاهر و حس و حال جدیدی به اندروید بخشیده است.

البته باید به این نکته توجه داشته باشید که چنین تغییراتی فقط به‌صورت ظاهری هستند و سیستم عامل اندروید این گوشی‌ها از بنیان و اساس یکسانی بهره می‌برند. با این وجود، اگر می‌خواهید هرچه سریع‌تر به‌روزرسانی‌های امنیتی و نرم‌افزاری گوگل را دریافت کنید، بهتر است که به استفاده از گوشی‌های مجهز به اندروید خالص، مانند گوگل پیکسل یا مجهز به اندروید وان روی بیاورید. چنین گوشی‌هایی قادر به دریافت مستقیم به‌روزرسانی‌های گوگل هستند. اما گوشی‌هایی که از اندرویدهای خالص بهره نمی‌برند، عموما چنین به‌روزرسانی‌هایی را خیلی دیرتر دریافت می‌کنند. دلیل این موضوع این است که شرکت‌های سازنده ابتدا باید این به‌روزرسانی‌ها را با توجه به اندروید گجت خود تطبیق دهند و بعد از انجام این کار و آزمایش‌های مربوطه، می‌توانند آن را ارائه کنند.

چنین موضوعی شاید از لحاظ امنیتی خبر خوبی نباشد، اما این رابط‌های کاربری مزیت‌هایی هم دارند. هرکدام از این رابط‌های کاربری ویژگی‌هایی را به ارمغان می‌آورند؛ ویژگی‌هایی که اندروید خالص از آن‌ها بی‌بهره است. به‌طور مثال می‌توانیم به اپلیکیشن مدیریت موبایل موجود در EMUI اشاره کنیم که در مقایسه با بیشتر اپلیکیشن‌های مشابه موجود در گوشی‌های دیگر، برتری محسوسی دارد. در همین زمینه نباید رابط کاربری گوشی‌های گلکسی سامسونگ را از قلم بیندازیم که می‌تواند عملکرد اپلیکیشن‌ها را با توجه به سخت‌افزار سامسونگ بهینه‌سازی کند. در نهایت انتخاب هرکدام از این رابط‌های کاربری به سلیقه‌ها و اولویت‌های شخصی کاربر برمی‌گردد و به همین خاطر توصیه می‌کنیم که قبل از خرید گوشی، چنین مواردی را در نظر بگیرید.

#### **سیستم‌عامل‌های مبتنی بر اندروید**



همانطور که پیش از این گفتیم، اندروید یک سیستم‌عامل متن‌باز و انعطاف‌پذیر است و به همین خاطر می‌توان تغییراتی را در آن اعمال کرد. سازندگان گوشی‌ها با بهره‌گیری از این قابلیت، هرکدام تغییرات موردنظر خود را بر اندروید اعمال می‌کنند. با وجود اینکه اندروید رایگان و متن‌باز است، اما سرویس‌های گوگل این‌گونه نیستند و سازندگان گوشی‌ها برای اینکه بتوانند از اپلیکیشن‌های موجود در پلی استور و سرویس‌های گوگل استفاده کنند، باید طبق قوانین این شرکت عمل کنند؛ موضوعی که بسیاری از شرکت‌ها آن را پذیرفته‌اند.

اما در این میان شرکت‌هایی هستند که نمی‌خواهند طبق قواعد گوگل عمل کنند. در اینجاست که سیستم‌عامل‌های مبتنی بر اندروید وارد عمل می‌شوند. مبنای چنین سیستم‌عامل‌هایی، اندروید است اما ارتباطی با اکوسیستم گوگل ندارند. از میان چنین سیستم‌عامل‌هایی، می‌توانیم به سیستم‌عامل تبلت «فایر» (Fire) شرکت آمازون اشاره کنیم. سیستم‌عامل این تبلت مبتنی بر اندروید بود، اما از هیچ‌کدام از اپلیکیشن‌ها یا سرویس‌های گوگل بهره نمی‌برد. برخی از شرکت‌های سازنده‌ی چینی کمتر شناخته‌شده هم از چنین سیستم‌عامل‌هایی استفاده می‌کنند؛ سیستم‌عامل‌هایی که به طرز شگفت‌آوری شبیه اندروید هستند، اما به پلی استور گوگل دسترسی ندارند.

# زبان های برنامه نویسی اندروید

**زبان Java جاوا**

انتخاب زبان Java انتخاب خوبی است. این زبان، زبان رسمی اندروید است. Java دومین زبان فعال در GitHub به حساب می‌آید و بیشتر از 20 سال سابقه دارد. تمام این حرف‌ها به این معناست که آموزش‌های فراوانی برای این زبان وجود دارد و به این زودی نگران از دور خارج شدن و منسوخ شدن این زبان نیستید. از آن جایی که Java در ده‌ها صنعت به کار گرفته می‌شود، ما به شما توصیه می‌کنیم از کتاب‌های آموزشی Java برای اندروید شروع کنید.

مشخصاً زبان Java در هر حال به همان شکل است اما الگوها و اصطلاح‌هایی وجود دارند که می‌توانند به داشتن درکی بهتر از برنامه نویسی در اندروید کمک کنند. این هدفی است که اکثر کتاب‌های اصولی اندروید در نظر دارند. به یاد داشته باشید از آنجایی که Java قدیمی است، برخی از خصوصیات و ظرافت‌های زبان‌های جدید را ندارد. اگر شما یک تازه‌کار هستید این موضوع بر شما تأثیر نمی‌گذارد اما اگر زبان‌های دیگر مانند Swift را بلد باشید، ممکن است در Java کمی احساس محدودیت به شما دست بدهد.

**۲. زبان Kotlin**

Kotlin یک زبان برنامه‌نویسی است که برای اجرا در ماشین مجازی Java طراحی شده است. این یعنی برنامه‌های Kotlin بر روی Bytecodeهای Java سوار هستند و به برنامه‌ها اجازه می‌دهد تا بر روی هر ماشینی که Java runtime Environment را پشتیبانی می‌کند، اجرا شوند. از آن جایی که اکثر ماشین‌ها این محیط را پشتیبانی می‌کنند، Kotlin یک راه به‌نسبت راحت برای خلق نرم‌افزارهای چندسکوییCross-platform است. با Kotlin شما بهترین خصوصیات Java را بدون نقاط ضعف آن در اختیار دارید. هماهنگی قسمت‌های مختلف و خصوصیات Kotlin امروزی، تمیز و سریع هستند و تجربه‌ برنامه‌نویسی راحتی را ارائه می‌کنند. در حالی که Java به نظر سنگین و منسوخ به نظر می‌رسد، Kotlin حس تازگی و هوشمندی دارد. درواقع به‌گونه‌ای می‌توانید به Kotlin به عنوان یک زبان طراحی‌شده برای نوشتن برنامه‌های اندروید فکر کنید.



نقطه ضعف Kotlin چیست؟ Kotlin بسیار جوان است. در سال 2011 ظهور پیدا کرد و تا سال 2016 به طور رسمی در اختیار همه نبود. خبر خوب این است که Kotlin رایگان و متن‌باز (Open Source) است، پس می‌توانید انتظار داشته باشید به سرعت پیشرفت کند اما چند سال طول می‌کشد تا یک انتخاب قابل اعتماد و مستحکم باشد.

**۳. زبان #C سی‌شارپ**

#C یک زبان خارق‌العاده است. به نظر من، #C شامل تمام بخش‌های خوب Java است، بدون نکات بد آن که حتی با بهینه‌سازی‌ها، بهتر هم شده است. شرکت مایکروسافت، پتانسیل Java را دید و تصمیم گرفت تا نسخه‌ بهتری از زبان برنامه‌نویسی خودش را تولید کند.
برای مدتی، بزرگ‌ترین اشکال #C این بود که تنها برای سیستم‌های ویندوزی اجرا می‌شد زیرا به NET Framework. متکی بود. به تدریج اما تمام اینها تغییر کرد. وقتی که مایکروسافت تصمیم گرفت NET Framework. را در سال 2014 به صورت متن‌باز در آورد و Xamarin شرکتی که برقراری Mono (پروژه‌ای که به #C اجازه می‌دهد تا بر روی بسیاری از بسترها اجرا شود را بر عهده دارد را خرید.

امروزه، شما می‌توانید از Xamarin.Android و Xamarin.IOS استفاده کنید تا با Visual Studio یا Xamarin Studio برنامه‌های موبایل بسازید. یادگیری #C انتخابی بسیار عالی است چون می‌توانید از این زبان در زمینه‌های دیگر، همچون ساختن بازی‌های پیچیده با Unity و #C استفاده کنید. یک مثال از برنامه‌های ساخته‌شده با Xamarin برنامه‌ی MarketWatch است. توجه داشته باشید که Xamarin در ابتدا یک برنامه‌ پولی بود اما شرکت مایکروسافت آن را به به یک برنامه‌ رایگان تبدیل کرد.

**۴.  زبان Pythonپایتون**

با این که اندروید از برنامه‌نویسی بومی Python پشتیبانی نمی‌کند، ابزارهایی وجود دارند که به شما اجازه می‌دهند در زبان Python برنامه بنویسید و بعد آن ها را به APK تبدیل کنید که به صورت موفقیت‌آمیز بر روی دستگاه‌های اندروید قابل اجرا هستند. این یک مثال از Python به عنوان زبان دنیای واقعی کاربردی است، این خبر خوبی برای کاربران Python است که می‌خواهند برنامه بسازند اما آن چنان از Java دل خوشی ندارند.



Kivy یکی از محبوب‌ترین و قدرتمندترین راه حل‌ها برای این کار است. نه‌تنها این برنامه به صورت متن‌باز است، نه‌تنها از ویندوز، Mac، لینوکس و iOS، جدای از اندروید، پشتیبانی می‌کند، بلکه به گونه‌ای طراحی شده است که شما را به برنامه نویسی سریع ترغیب کند. شما می‌توانید از آن به عنوان ابزار نمونه‌سازی استفاده کنید و این کار را با تنها چند خط کد انجام دهید.

از آن جایی که این برنامه به صورت native پشتیبانی نمی‌شود، نمی‌توانید از فواید native آن بهره‌مند شوید. برنامه‌هایی که با Kivy طراحی می‌شوند، معمولاً APK‌های بزرگ‌تری دارند، کندتر شروع می‌شوند و کارایی آن‌ها کمتر از حد مورد انتظار است. با این وجود هر نسخه جدید از قبلی بهتر است و دستگاه‌ها به حدی قوی شده‌اند که این موضوع آن‌چنان اهمیت ندارد، پس نگذارید این مساله جلوی کار شما را بگیرد.

**۵. زبان‌های HTML5 + CSS + JavaScript**

این سه زبان که آغاز کارشان با عنوان هسته‌ سه‌گانه‌ عالی برای برنامه‌نویسی front-end بود، از آن زمان تکامل پیدا کردند و حتی به چیزی بهتر رسیده‌اند. هم‌اکنون می‌توانید با آن‌ها هر نوع برنامه‌ای بسازید، چه برای گوشی و چه برای رایانه. در اصل شما دارید یک اپلیکیشن تحت وب (Web app) می‌سازید که به عنوان یک برنامه‌ آفلاین در بستر جادویی رندر می‌شود.

برای ساختن برنامه‌های اندروید از این طریق، می‌توانید از Adobe Cordova استفاده کنید که یک چارچوب متن‌باز است و iOS، ویندوز 10 موبایل، Blackberry ،Firefox و… را پوشش می‌دهد. Cordova با وجود کارایی بالایی که دارد، نیازمند تلاش زیادی برای خلق و اجرای یک برنامه‌ نسبتاً ساده است. برای همین بسیاری افراد از Ionic Framework (که از Cordova برای پیاده‌سازی در بسترهای گوناگون بهره می‌گیرد.) استفاده می‌کنند. مثال‌هایی از این مورد برای برنامه‌های اندرویدی عبارت‌اند از برنامه‌های Untappd و TripCase.

یک گزینه‌ دیگر این است که از React Native استفاده کنید. این کتابخانه می‌تواند در اندروید، iOS، و بستر جهانی ویندوز قرار گیرد. توسط فیسبوک، اینستاگرام و سایر شرکت‌های بزرگ استفاده و نگه‌داری می‌شود. پس می‌توانید مطمئن باشید که امتحان و تست شده است. یادگیری آن آسان نیست اما به محض آن که از مسائل اولیه بگذرید، تمام قدرت و انعطاف‌پذیری را که می‌خواهید، به دست می‌آورید.

**. 6 زبان Lua**

Lua یک زبان نوشتاری قدیمی است که در اصل طراحی شده تا برنامه‌های نوشته‌شده در زبان‌های قدرتمندتر مانند C ،VB.net و … را تکمیل کند. بر همین اساس، دارای برخی خصوصیات است که آن را با دیگران متفاوت می‌کند؛ مانند داشتن آرایه‌ ابتدایی (Starting array) در یک به جای صفر و نداشتن کلاس‌های native.

جدای از این صحبت‌ها، در برخی موارد می‌توان از Lua به عنوان یک زبان برنامه‌نویسی اصلی استفاده کرد و Corona SDK یک مثال عالی است. با Corona می‌توانید برنامه‌هایی غنی ایجاد کنید که از Lua استفاده می‌کنند و می‌توان آن‌ها را بر ویندوز، Mac، اندروید، iOS و حتی Apple TV و Android TV قرار داد. همچنین در آن خصوصیت کسب درآمد به علاوه‌ یک محل بازار گسترده دارد که در آن می‌توانید ویژگی‌ها و افزونه (Plugin) بخرید.
از Corona عمدتاً برای ساختن بازی استفاده می‌شودمانند Fun Run2 و HoPiko اما می‌توان از آن برای ساختن برنامه‌های خدمات رفاهی و تجاریمانند My Days و Quebec Tourism استفاده کرد.

**۷. زبان‌های C و ++C**

گوگل برای ساختن برنامه‌های اندرویدی، دو ابزار برنامه نویسی ارائه می‌دهد: SDK که از Java استفاده می‌کند و NDK که از زبان‌های بومی مانند C و ++C استفاده می‌کند. به یاد داشته باشید که نمی‌توانید تمام یک برنامه را با استفاده از C++ ،C و Java بسازید، بلکه زبان NDK به شما اجازه می‌دهد تا یک کتابخانه بومی ایجاد کنید و قابلیت‌هایش را در کد Java به کار ببرید.

در اکثر مواقع نیازی به استفاده از NDK نخواهید داشت. نباید از آن فقط به این دلیل که ترجیح می‌دهید به جای Java در C++/C کد بزنید استفاده کنید، بلکه زبان NDK برای زمانی است که می‌خواهید به زور از وظایف حسابگری سنگین، کارایی بیشتری بیرون بکشید. همچنین به شما این امکان را می‌دهد تا در برنامه‌ خود کتابخانه‌های C و یا ++C جای دهید.

اما در غیر این صورت، باید هرگاه ممکن بود به Java تکیه کنید. برنامه‌نویسی با استفاده از C++/C بسیار پیچیده‌تر از Java است و برخی اوقات چیز زیادی از آن نمی‌گیرید.

# ویژگی های زبان کاتلین چیست؟

مدتی است که کاتلین به نقل محفل برنامه نویسان اندروید و Trend روز برنامه نویسان ایرانی و جهانی تبدیل شده است. در حال حاضر کمتر وب سایتی را مشاهده خواهید کرد که [آموزش کاتلین](https://learnfiles.com/course/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D8%B2%D8%A8%D8%A7%D9%86-%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87-%D9%86%D9%88%DB%8C%D8%B3%DB%8C-%DA%A9%D8%A7%D8%AA%D9%84%DB%8C%D9%86/) و یا معرفی آن را به گونه ای پوشش نداده باشد . در این مقاله به معرفی جامعی از کاتلین و ویژگی های آن می پردازیم همچنین شرکت های معروفی که از این زبان بهره برده اند را معرفی خواهیم کرد.

کاتلین در سال ۲۰۱۱ توسط [JetBrains](https://www.jetbrains.com/) معرفی شد و در همان تاریخ هم با استقبال خوبی از طرف برنامه نویسان مواجه شد ولی در کنفرانس [IO17 گوگل](https://events.google.com/io2017/) بود که گوگل رسما از این زبان پشتیبانی کرد و آن را به Trend روز جهان و همچنین ایران در حوزه برنامه نویسی تبدیل کرد.

از بدو تولد اندروید معضلی که همه برنامه نویسان اندروید رو آزرده خاطر کرده بود استفاده از یک زبان قدیمی (جاوا) و از محیط برنامه نویسی قدیمی (IDE : Eclipse) برای توسعه اپلیکیشن های اندرویدی بود که بعدا با معرفی اندروید استودیو مقداری از این آزرده خاطری کاهش پیدا کرد ولی هنوز پابرجا بود.

**کاتلین چیست ؟**

کاتلین یک زبان برنامه نویسی ایستا (Statically typed) که برای چهار هدف قابل استفاده است :

* برای ماشین مجازی جاوا(JVM)
* برای توسعه اپلیکیشن های اندرویدی
* برای مرورگر ها به کمک جاوااسکریپت
* برای Kotlin/Native

کنفرانس IO17 گوگل رسما از کاتلین پشتیبانی کرد و آن را زبانی جدید برای توسعه اپلیکیشن های اندروید قرار داد، کاتلین بیشتر برای توسعه اپلیکیشن های اندروید شناخته شد. در حقیقت از این تاریخ به بعد بود که [Kotlin](https://kotlinlang.org/) توجه بیشتری از برنامه نویسان به خود جلب کرد. البته این جلب توجه غیرمنتظره نبود زیرا کاتلین زبانی ذاتا کامپایل شده است و سریع تر از زبان های دیگر عمل میکند و در نتیجه اپلیکیشن هایی بهتر و قوی تری میتوان با آن تولید کرد . از همه مهم تر اینکه این زبان با زبان های پر استفاده دیگر اندروید مانند جاوا کاملا سازگار است.

کاتلین یک زبان منبع باز محسوب میشود که ویژگی های برنامه نویسی شی گرا و ماژولار را با هم ترکیب کرده است  . نام کاتلین از جزیره ای به همین نام در اطراف سن پترزبورگ روسیه گرفته شده است . به این دلیل که بیشتر کارهای توسعه Kotlin در شعبه سن پترزبورگ JetBrains انجام شد.

**چرا کاتلین ؟**

اولین دلیلی که میتوان برای استفاده از کاتلین نام برد سازگاری کامل کاتلین با جاواست . شما در هر جای پروژه ای که با جاوا نوشته اید میتوانید از کدهای کاتلین هم استفاده کنید و نگران هیچ مشکلی مانند افت سرعت اجرای پروژه خود نباشید . از طرف دیگر شما میتوانید کدهای جاوا را به کاتلین و بالعکس تبدیل کنید. هر مقدار از پروژه ای که به زبان جاوا نوشته اید میتوانید به راحتی به زبان کاتلین تبدیل کنید و از ویژگی های زبان کاتلین به طور کامل استفاده کنید.

سه مزیت و ویژگی مهم زبان کاتلین که برنامه نویسان را از زبان جاوا به زبان کاتلین راغب می‌کند:

* ۱. اولین دلیل قدیمی بودن زبان جاواست که قدمتی ۲۲ ساله دارد و شاید خیلی از تکنولوژی های جدید را نتوان و یا به سختی و پیچیدگی فراوان بتوان با آن پیاده سازی کرد.
* ۲. جاوا ترکیبی از کدهای طولانی است.
* ۳. کامپایل کدهای جاوا بسیار کند است.
* ۴. همچنین قابلیت برنامه نویسی End-To-End کاتلین نیز وجود دارد که امکان استفاده از آن در بسیاری از پلتفرم ها را فراهم میکند

**از کاتلین در کجاها می توان استفاده کرد؟!**

از کوتلین می توانید برای هر نوع برنامه نویسی چه در سمت سرور و client و چه اندروید استفاده کنید. با کمک Kotlin/Native شما می توانید از پشتیبانی مک و IOS نیز بزودی بهره مند شوید. بیشتر افراد از کوتلین برای موبایل و تحت وب استفاده می کنند و برای برنامه نویسی سمت Client از جاوا اسکریپت و جاوا Fx استفاده می شود. Data science نیز جزو گزینه های موجود است.

**Kotlin برای توسعه سرور**

کاتلین سازگار با JVM است و به همین علت می توانید از هر فریم ورک موجود مانند Spring Boot، vert.x یا JSF استفاده کنید. علاوه بر این، فریم ورک خاصی در کوتلین مانند Ktor وجود دارد.

**Kotlin برای توسعه وب**

شما می توانید از  kotlin js نه تنها برای کدنویسی در سمت سرور بلکه برای client نیز استفاده کنید . کمترین از تعاریف موجود در  definitely Typedاستفاده می کند تا دستور های رایج در کتابخانه های جاوا را اجرا کند. ابن برنامه با سیستم های ماژولی مپل AMDو  common js  نیز هم خوانی دارد.

**Kotlin برای توسعه دسکتاپ**

 همچنین شما می توانید از هر فریم ورک رابط کاربری جاوا مانند JavaFx، Swing یا دیگر استفاده کنید. علاوه بر این، فریم ورک خاص Kotlin مانند TornadoFX وجود دارد.

**Kotlin برای توسعه بومی (Kotlin / Native)**

Kotlin / Native در حال حاضر در حال بررسی است. Kotlin / Native کاتلین را به کد های  native کامپایل می کند که این کدها می تواند بدون VM (ماشین مجازی) اجرا شود. پیشن مایش این فناوری منتشر شده است اما هنوز آماده تولید نیست و در حال حاضر تمام سیستم عاملها برای پشتیبانی از آن مورد هدف نیستند.

**Kotlin برای توسعه اندروید**

Kotlinبه عنوان یک زبان درجه اول در اندروید پشتیبانی می شود. صدها برنامه کاربردی که از Kotlin برای اندروید استفاده می کنند که به چند مورد از آن‌ها اشاره می کنیم:

**تعدادی از برنامه هایی که با کاتلین نوشته شده اند:**

**Pinterest**

سرویس اجتماعی معروف اشتراک عکس تصمیم گرفت از Kotlin استفاده کند و در برنامه Android خود از جاوا بهره ببرد. آنها مزایا و معایب استفاده از Kotlin را در تولید در Droidcon NYC 2016 ارائه دادند. چیزی که ممکن است کمی عجیب باشد، این است که Kotlin در چنین برنامه محبوب با پایگاه کاربری گسترده اش حتی قبل از اینکه گوگل اولین پشتیبانی از این زبان را اعلام کند، استفاده کرد.

**Coursera**

Coursera یک شرکت آموزشی است که دوره های آموزشی آنلاین را که توسط دانشگاه ها و موسسات برجسته مانند استنفورد و ییل برگزار می شود، ارائه می دهد. Coursera سازمان های Android و iOS جداگانه ندارد. آنها فقط مفهوم تلفن همراه را دارند و به همین دلیل تصمیم گرفتند از Kotlin استفاده کنند. Kotlin شبیه Swift است، یک زبان برنامه نویسی نسبتا جدید برای توسعه iOS است که این موضوع باعث ساده تر به اشتراک گذاری دانش بین مهندسان، ارتباطات بهتر و توسعه کلی کارآمدتر می شود.

**Trello**

Trelloیک برنامه مدیریت شناخته شده پروژه است که در حال حاضر متعلق به Atlassian است. ترلو برای برنامه ی Android خود، کد های جدید ش را در Kotlin نوشته است.همکاری با Java و Kotlin ممکن است به لطف قابلیت همکاری عالی بین دو زبان باشد.

**Evernote**

Evernote یک برنامه طراحی شده برای ضبط، سازماندهی، لیست کارها و بایگانی است. Evernote  برای مشتریان اندروید خود از زبان کاتلین بهره برده است.

**Basecamp 3**

برنامه ای که به مدیریت کسب و کارهای کوچک در حال رشد کمک می کند. این به طور کامل در Kotlin نوشته شده است و توسعه دهندگان Basecamp ادعا می کنند که انتخاب Kotlin منجر به تفاوت های بزرگی در آسایش، کیفیت و سرعت کار آنها شد.

**Twidere برای توییتر**

منبع باز، ۱۰۰ درصد طراحی مرورگر توییتر با material design، افزایش تجربه کاربر توییتر. همه‌ی  این ها  در کوتلین نوشته شده است، اگر چه برخی از کدهای جاوا هنوز در آن وجود دارد.

**Shadowsocks**

یک cross-platform با عملکرد و امنیت بالا است که از Proxy Socks5 پشتیبانی کرده و کمک می کند تا در هنگام مرور اینترنت افزایش امنیت و حریم خصوصی داشته باشید. Shadowsocks از زبان کاتلین برای مشتریان اندرویدی خود بهره برده است. Shadowsocks هر دو توسط کاربران در Google Play و سایر توسعه دهندگان قدردانی می شود.

**Simple Calendar**

یک تقویم ساده و Open source بدون نیاز به مجوز های خاص. این برنامه، جزئی از پک کامل موبایلی Simple Mobile Tools می باشد. همه‌ی این برنامه ها Open Source بوده و در کوتلین نوشته شدند.

**برنامه KotlinConf**

KotlinConf یک کنفرانس توسعه دهندگان است که توسط سازندگان کوتلین JetBrains میزبانی شده است. هیچ راهی وجود ندارد که برنامه کنفرانس رسمی در یک زبان دیگر نوشته شود. واقعا جالب است این است که نه تنها برای مشتریان Android این برنامه در Kotlin نوشته شده بلکه برنامه وب، سرور و IOS نیز دارد. این فقط نشان دهنده پتانسیل عالی این زبان است.

**برنامه رسمی کنفرانس Driodkaigi 2018**

DroidKaigi 2018  یک کنفرانس برای برنامه نویسان اندرویدی است که در توکیو سازمان دهی می شود. این یکی  دیگر از برنامه هایی است که توسط کوتلین نوشته شده ولی این بار کنفرانس توسط خلاقان این برنامه برگزار نمی شود. این موضوع نشان دهنده ی محبوبیت بالای کوتلین در بین اندرویدی هاست

**IDEهای پشتیبانی شده از KOTLIN کدام‌اند؟**

Kotlinتوسط تمام IDE های اصلی Java شامل IntelliJ IDEA، Android Studio، Eclipse و NetBeans پشتیبانی می شود. علاوه بر این، کامپایلر خط فرمان در دسترس است و پشتیبانی مستقیم برای کامپایل و اجرای برنامه ها را فراهم می کند.

**ساخت ابزارهای پشتیبانی شده توسط**[**کوتلین**](https://learnfiles.com/blog/%D8%B2%D8%A8%D8%A7%D9%86-%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87-%D9%86%D9%88%DB%8C%D8%B3%DB%8C-%DA%A9%D9%88%D8%AA%D9%84%DB%8C%D9%86/)

در سمت JVM ابزار اصلی ساخت شامل Gradle، Maven، Ant و Kobalt است. همچنین برخی از ابزارهای ساخت موجود وجود دارد که جاوا اسکریپت سمت کلاینت را هدف قرار می دهند.

**کوتلین یا همان کاتلین چه چیزی را تشکیل می دهد؟**

هنگامی که JVM را هدف قرار می دهد، کوتلین بایت کد های سازگار با جاوا را تولید می کند. هنگامی که جاوا اسکریپت را هدف قرار می دهد، کوتلین به ES5.1 رجوع می کند و کد را تولید می کند که سازگار با سیستم های ماژول از جمله AMD و CommonJS است. وقتی کوتلین بومی (Kotlin / Native) را هدف قرار می‌دهد، کد مخصوص پلتفرم را تولید می کند (از طریق LLVM).

**آیا کوتلین فقط جاوا ۶ را هدف قرار می دهد؟**

 Kotlin به شما اجازه می دهد بین تولید کننده های Java 6 و  Java 8 کدهای سازگار با نسخه موردنظر را انتخاب کنید. اکثر کد های بهینه  ممکن است برای نسخه های بالاتر پلتفرم تولید شود.

**کدام شرکت ها از KOTLIN استفاده می کنند؟**شرکت های زیادی وجود دارند که از Kotlin استفاده می کنند، اما برخی از شرکت های مشهور دیگری که به طور عمومی استفاده از Kotlin اعلام کرده اند، از طریق پست های وبلاگ، مخازن GitHub یا مذاکرات شامل Square، Pinterest یا Basecamp است.

# آماده سازی ابزار ها و محیط

**نصب jdk**

برای شروع برنامه نویسی ابتدا باید محیط و لوازم برنامه نویس را فراهم کنیم

برای اینکار ابتدا باید jdk را نصب کنیم.این ابزار اجازه می دهد که ما برنامه های به زبان جاوا را اجرا کنیم .

**نصب android stadio**

اخرین نسخه android stadio را می توانیم از سایت خود android stadio دانلود کنیم . هنگام نصب خود اندروید استدیو sdk , gradl و دیگر ابزار های لازم را دانلود و نصب می کند .بعد از نصب هم می توانیم سایر ابزارها و sdk های مورد نیاز را نصب کنیم

**راه اندازی ماشین مجازی**

 برای اجرای برنامه ها از ماشین مجازی استفاده می کنیم که محیط سیستم عامل اندروید را برای ما شبیه سازی می کند .

**ثبت نام درfirebase و ایجاد پروژه**

برای ثبت نام باید از ابزار تغیر ip استفاده کنید. Friebase یک سیستم رایگانه ولی بعضی خدماتش غیر رایگان است وباید براش هزینه کنید .

امکانات زیادی این سیستم برای ما فراهم میکند . ما برای زخیزه اطلاعات و ارسال پیام از firebase استفاده می کنیم .

من برای ساخت این پیام رسان از 5 اکتیویتی استفاده می کنم ثبت نام لاگین ، صفحه چت ، صفحه پیام ها ، صفحه کاربران

فایل مانیفست

فایل مانیفست یک فایل مهم است که اکتیویتی و اسم برنامه ویژگی های برنامه را مشخص می کند .

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
 package="com.example.mymasenger">

 <application
 android:allowBackup="true"
 android:icon="@mipmap/ic\_launcher"
 android:label="@string/app\_name"
 android:roundIcon="@mipmap/ic\_launcher\_round"
 android:supportsRtl="true"
 android:theme="@style/AppTheme">
 <activity android:name=".LastMassegeActivity">
 <intent-filter>
 <action android:name="android.intent.action.MAIN" />

 <category android:name="android.intent.category.LAUNCHER" />
 </intent-filter>
 </activity>
 <activity android:name=".MainActivity"/>
 <activity android:name=".Login" />
 <activity android:name=".NewmassegeActivity">
 <meta-data
 android:name="android.support.PARENT\_ACTIVITY"
 android:value=".LastMassegeActivity" />
 </activity>

 <activity android:name=".ChatLogActivity" >
 <meta-data
 android:name="android.support.PARENT\_ACTIVITY"
 android:value=".LastMassegeActivity" />

 </activity>
 </application>

</manifest>

کد های رابط کاربری

Activity\_main

این اکتیوتی صفحه صفحه ثبت نام را ایجاد می کند. که از 3 edit text برای دریافت اسم کاربر و پسورد و ایمیل استفاده می کنیم .از یک image view , رای عکس پروفایل و یک Button برای ثبت نام و text view برای برای برای رفتن به صفحه لاگین استفاده می کنیم و به هریک ویژگی های انها را می دهیم .

<androidx.constraintlayout.widget.ConstraintLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 xmlns:tools="http://schemas.android.com/tools"
 android:layout\_width="match\_parent"
 android:layout\_height="match\_parent"
 android:background= "#8decff"
 tools:context=".MainActivity">

 <EditText
 android:id="@+id/username\_edittext\_register"
 android:layout\_width="0dp"
 android:layout\_height="50dp"
 android:layout\_marginEnd="32dp"
 android:layout\_marginLeft="32dp"
 android:layout\_marginRight="32dp"
 android:layout\_marginStart="32dp"
 android:layout\_marginTop="32dp"
 android:textColorHint="@android:color/black"
 android:textColor="@android:color/black"
 android:background="@drawable/rounded\_edittext\_register\_login"
 android:ems="10"
 android:hint="Username"
 android:inputType="textPersonName"
 android:paddingLeft="16dp"
 app:layout\_constraintEnd\_toEndOf="parent"
 app:layout\_constraintStart\_toStartOf="parent"
 app:layout\_constraintTop\_toBottomOf="@+id/selectphoto\_button\_register" />

 <EditText
 android:id="@+id/email\_edittext\_register"
 android:layout\_width="0dp"
 android:layout\_height="50dp"
 android:layout\_marginTop="8dp"
 android:textColor="@android:color/black"
 android:background="@drawable/rounded\_edittext\_register\_login"
 android:ems="10"
 android:hint="Email"
 android:textColorHint="@android:color/black"
 android:inputType="textEmailAddress"
 android:paddingLeft="16dp"
 app:layout\_constraintEnd\_toEndOf="@+id/username\_edittext\_register"
 app:layout\_constraintStart\_toStartOf="@+id/username\_edittext\_register"
 app:layout\_constraintTop\_toBottomOf="@+id/username\_edittext\_register" />

 <EditText
 android:id="@+id/password\_edittext\_register"
 android:layout\_width="0dp"
 android:layout\_height="50dp"
 android:layout\_marginTop="8dp"
 android:background="@drawable/rounded\_edittext\_register\_login"
 android:ems="10"
 android:hint="Password"
 android:textColorHint="@android:color/black"
 android:inputType="textPassword"
 android:textColor="@android:color/black"
 android:paddingLeft="16dp"
 app:layout\_constraintEnd\_toEndOf="@+id/email\_edittext\_register"
 app:layout\_constraintStart\_toStartOf="@+id/email\_edittext\_register"
 app:layout\_constraintTop\_toBottomOf="@+id/email\_edittext\_register" />

 <Button
 android:id="@+id/register\_button\_register"
 android:layout\_width="0dp"
 android:layout\_height="wrap\_content"
 android:layout\_marginTop="8dp"
 android:background="@drawable/rounded\_button"
 android:text="Register"

 android:textColor="@android:color/white"
 android:textStyle="bold"
 app:layout\_constraintEnd\_toEndOf="@+id/password\_edittext\_register"
 app:layout\_constraintStart\_toStartOf="@+id/password\_edittext\_register"
 app:layout\_constraintTop\_toBottomOf="@+id/password\_edittext\_register" />

 <TextView
 android:id="@+id/already\_have\_account\_text\_view"
 android:layout\_width="wrap\_content"
 android:layout\_height="wrap\_content"
 android:layout\_marginEnd="8dp"
 android:layout\_marginLeft="8dp"
 android:layout\_marginRight="8dp"
 android:layout\_marginStart="8dp"
 android:layout\_marginTop="8dp"
 android:text="Already have an account?"
 android:textColor="@android:color/white"
 android:textSize="16sp"
 android:textStyle="bold"
 app:layout\_constraintEnd\_toEndOf="@+id/register\_button\_register"
 app:layout\_constraintStart\_toStartOf="@+id/register\_button\_register"
 app:layout\_constraintTop\_toBottomOf="@+id/register\_button\_register" />

 <Button
 android:id="@+id/selectphoto\_button\_register"
 android:layout\_width="150dp"
 android:layout\_height="150dp"
 android:layout\_marginEnd="8dp"
 android:layout\_marginLeft="8dp"
 android:layout\_marginRight="8dp"
 android:layout\_marginStart="8dp"
 android:layout\_marginTop="32dp"
 android:background="@drawable/rounded\_select\_photo\_button"
 android:text="Select Photo"
 android:textColor="@android:color/white"
 android:textStyle="bold"
 app:layout\_constraintEnd\_toEndOf="parent"
 app:layout\_constraintStart\_toStartOf="parent"
 app:layout\_constraintTop\_toTopOf="parent" />

 <de.hdodenhof.circleimageview.CircleImageView
 android:id="@+id/selectphoto\_imageview\_register"
 app:civ\_border\_width="4dp"
 android:layout\_width="150dp"
 android:layout\_height="150dp"
 app:layout\_constraintBottom\_toBottomOf="@+id/selectphoto\_button\_register"
 app:layout\_constraintEnd\_toEndOf="@+id/selectphoto\_button\_register"
 app:layout\_constraintStart\_toStartOf="@+id/selectphoto\_button\_register"
 app:layout\_constraintTop\_toTopOf="@+id/selectphoto\_button\_register" />

</androidx.constraintlayout.widget.ConstraintLayout>

Last\_massege\_activity

این اکتویتی برای نمایش پیام های کاربران استفاده می کنیم RecyclerView یک لیست برای نمایش پیام ها ایجاد می کند

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 xmlns:tools="http://schemas.android.com/tools"
 android:layout\_width="match\_parent"
 android:layout\_height="match\_parent"
 android:background="#8decff"
 tools:context=".MainActivity">

 <androidx.recyclerview.widget.RecyclerView
 android:id="@+id/recyclerview\_latest\_messages"
 android:layout\_width="0dp"
 android:layout\_height="0dp"
 app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
 app:layout\_constraintBottom\_toBottomOf="parent"
 app:layout\_constraintEnd\_toEndOf="parent"
 app:layout\_constraintStart\_toStartOf="parent"
 app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

Last\_massege\_row

این layout برای برای تنظیم ظاهر ردیف هر یک از ایتم های لیست بالا استفاده می کنیم یک imageview برای نمایش عکس پروفایل و دوتا textview برای نمایش اسم و قسمتی از پیام ارسال شده یا پیام دریافت شده

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:layout\_width="match\_parent"
 android:layout\_height="120dp"

 xmlns:app="http://schemas.android.com/apk/res-auto"
 xmlns:tools="http://schemas.android.com/tools">

 <de.hdodenhof.circleimageview.CircleImageView
 android:id="@+id/imageview\_latest\_message"
 android:layout\_width="72dp"
 android:layout\_height="72dp"
 android:layout\_marginBottom="8dp"
 android:layout\_marginLeft="24dp"
 android:layout\_marginStart="24dp"
 android:layout\_marginTop="8dp"
 app:civ\_border\_width="4dp"
 app:layout\_constraintBottom\_toBottomOf="parent"
 app:layout\_constraintStart\_toStartOf="parent"
 app:layout\_constraintTop\_toTopOf="parent"
 app:srcCompat="@android:drawable/btn\_star\_big\_on" />

 <TextView
 android:id="@+id/username\_textview\_latest\_message"
 android:layout\_width="wrap\_content"
 android:layout\_height="wrap\_content"
 android:layout\_marginLeft="16dp"
 android:layout\_marginStart="16dp"
 tools:text="Username"
 android:textColor="@android:color/black"
 android:textStyle="bold"
 android:textSize="16sp"
 app:layout\_constraintBottom\_toTopOf="@+id/message\_textview\_latest\_message"
 app:layout\_constraintStart\_toEndOf="@+id/imageview\_latest\_message"
 app:layout\_constraintTop\_toTopOf="parent"
 app:layout\_constraintVertical\_chainStyle="packed" />

 <TextView
 android:id="@+id/message\_textview\_latest\_message"
 android:layout\_width="0dp"
 android:layout\_height="wrap\_content"
 android:layout\_marginEnd="16dp"
 android:layout\_marginLeft="16dp"
 android:layout\_marginRight="16dp"
 android:layout\_marginStart="16dp"
 android:layout\_marginTop="8dp"
 android:text="Latest Message"
 android:textSize="18sp"
 android:textColor="@android:color/black"
 app:layout\_constraintBottom\_toBottomOf="parent"
 app:layout\_constraintEnd\_toEndOf="parent"
 app:layout\_constraintHorizontal\_bias="0.5"
 app:layout\_constraintStart\_toEndOf="@+id/imageview\_latest\_message"
 app:layout\_constraintTop\_toBottomOf="@+id/username\_textview\_latest\_message" />
</androidx.constraintlayout.widget.ConstraintLayout>

Activity\_newmassege

این layout برای نمایش کاربران برای انتخاب یک کاربر بای ارسال پیام می باشد

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 xmlns:tools="http://schemas.android.com/tools"
 android:layout\_width="match\_parent"
 android:layout\_height="match\_parent"
 android:background="#8decff"
 tools:context=".MainActivity">

 <androidx.recyclerview.widget.RecyclerView
 android:id="@+id/recyclerview\_newmessage"
 android:layout\_width="0dp"
 android:layout\_height="0dp"
 app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
 app:layout\_constraintBottom\_toBottomOf="parent"
 app:layout\_constraintEnd\_toEndOf="parent"
 app:layout\_constraintStart\_toStartOf="parent"
 app:layout\_constraintTop\_toTopOf="parent"
 />
</androidx.constraintlayout.widget.ConstraintLayout>

Last\_massege\_row

این layout هر یک از ردیف های لیست بالا را تنظیم می کند .

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 xmlns:tools="http://schemas.android.com/tools"
 android:layout\_width="match\_parent"
 android:layout\_height="wrap\_content">

 <TextView
 android:id="@+id/username\_textview\_new\_message"
 android:layout\_width="0dp"
 android:layout\_height="wrap\_content"
 android:layout\_marginBottom="8dp"
 android:layout\_marginEnd="8dp"
 android:layout\_marginLeft="16dp"
 android:layout\_marginRight="8dp"
 android:layout\_marginStart="16dp"
 android:text="Username"
 android:textColor="@android:color/black"
 android:textSize="16sp"
 android:textStyle="bold"
 app:layout\_constraintBottom\_toBottomOf="@+id/imageview\_new\_message"
 app:layout\_constraintEnd\_toEndOf="parent"
 app:layout\_constraintStart\_toEndOf="@+id/imageview\_new\_message"
 app:layout\_constraintTop\_toTopOf="@+id/imageview\_new\_message" />

 <de.hdodenhof.circleimageview.CircleImageView
 android:id="@+id/imageview\_new\_message"
 android:layout\_width="60dp"
 android:layout\_height="60dp"
 android:layout\_marginLeft="16dp"
 android:layout\_marginStart="16dp"
 android:layout\_marginTop="16dp"
 app:civ\_border\_width="2dp"
 app:layout\_constraintStart\_toStartOf="parent"
 app:layout\_constraintTop\_toTopOf="parent"
 app:srcCompat="@android:drawable/btn\_dialog" />

</androidx.constraintlayout.widget.ConstraintLayout>

Activity\_chat\_log

این layout برای صفحه چت است که شامل یک لیست چت وtext view و یک button است

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 xmlns:tools="http://schemas.android.com/tools"
 android:layout\_width="match\_parent"
 android:layout\_height="match\_parent"
 android:background="#8decff"
 tools:context=".ChatLogActivity">

 <EditText
 android:id="@+id/edittext\_chat\_log"
 android:layout\_width="0dp"
 android:layout\_height="50dp"
 android:layout\_marginBottom="8dp"
 android:layout\_marginEnd="8dp"
 android:layout\_marginLeft="8dp"
 android:layout\_marginRight="8dp"
 android:textColor="@android:color/black"
 android:layout\_marginStart="8dp"
 android:ems="10"
 android:textColorHint="@android:color/black"
 android:textAlignment="center"
 android:hint="Enter Message"
 android:background="@drawable/rounded\_edittext\_register\_login"
 android:inputType="textPersonName"
 app:layout\_constraintBottom\_toBottomOf="parent"
 app:layout\_constraintEnd\_toStartOf="@+id/send\_button\_chat\_log"
 app:layout\_constraintStart\_toStartOf="parent"/>
 <Button
 android:id="@+id/send\_button\_chat\_log"
 android:layout\_width="wrap\_content"
 android:layout\_height="50dp"
 android:layout\_marginBottom="8dp"
 android:layout\_marginEnd="8dp"
 android:layout\_marginRight="8dp"
 android:textStyle="bold"
 android:text="send"
 android:textColor="@android:color/black"
 android:background="@drawable/rounded\_edittext\_register\_login"
 app:layout\_constraintBottom\_toBottomOf="parent"
 app:layout\_constraintEnd\_toEndOf="parent" />

 <androidx.recyclerview.widget.RecyclerView
 android:id="@+id/recyclerview\_chat\_log"
 android:layout\_width="0dp"
 android:layout\_height="0dp"
 android:layout\_marginBottom="8dp"
 app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
 app:layout\_constraintBottom\_toTopOf="@+id/edittext\_chat\_log"

 app:layout\_constraintEnd\_toEndOf="parent"
 app:layout\_constraintStart\_toStartOf="parent"
 app:layout\_constraintTop\_toTopOf="parent"
 />
</androidx.constraintlayout.widget.ConstraintLayout>

Chat\_row\_to

این layout برای تنظیم هر ایتم پیام ارسالی است

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 android:layout\_width="match\_parent"
 android:layout\_height="wrap\_content"
 android:layout\_marginBottom="12dp">
 <TextView
 android:id="@+id/textview\_to\_row"
 android:layout\_width="wrap\_content"
 android:layout\_height="wrap\_content"
 android:layout\_marginEnd="8dp"
 android:layout\_marginRight="8dp"
 android:background="@drawable/from\_to"
 android:maxWidth="240dp"
 android:padding="16dp"
 android:textColor="@android:color/black"

 android:text="This is my message that will wrap into multiple lines and keep on going"
 app:layout\_constraintEnd\_toStartOf="@+id/imageview\_chat\_to\_row"
 app:layout\_constraintTop\_toTopOf="@+id/imageview\_chat\_to\_row" />

 <de.hdodenhof.circleimageview.CircleImageView
 android:id="@+id/imageview\_chat\_to\_row"
 android:layout\_width="50dp"
 android:layout\_height="50dp"
 android:layout\_marginEnd="8dp"
 android:layout\_marginRight="8dp"
 android:layout\_marginTop="8dp"
 app:civ\_border\_width="2dp"
 app:civ\_border\_color="@android:color/black"
 app:layout\_constraintEnd\_toEndOf="parent"
 app:layout\_constraintTop\_toTopOf="parent"
 app:srcCompat="@android:drawable/btn\_star\_big\_on" />

</androidx.constraintlayout.widget.ConstraintLayout>

Chat\_row\_from

این layout برای نمایش پیام های دریافتی در از کاربران است

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:layout\_width="match\_parent"
 android:layout\_height="wrap\_content"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 android:layout\_marginBottom="12dp">
 <TextView
 android:id="@+id/textview\_from\_row"
 android:layout\_width="wrap\_content"
 android:layout\_height="wrap\_content"
 android:layout\_marginLeft="8dp"
 android:layout\_marginStart="8dp"
 android:text="This is my message that will wrap into multiple lines and keep on going"
 android:maxWidth="240dp"
 android:background="@drawable/to"
 android:padding="16dp"
 android:textColor="@android:color/black"
 app:layout\_constraintStart\_toEndOf="@+id/imageview\_chat\_from\_row"
 app:layout\_constraintTop\_toTopOf="@+id/imageview\_chat\_from\_row" />

 <de.hdodenhof.circleimageview.CircleImageView
 android:id="@+id/imageview\_chat\_from\_row"
 android:layout\_width="50dp"
 android:layout\_height="50dp"
 android:layout\_marginLeft="8dp"
 android:layout\_marginStart="8dp"
 android:layout\_marginTop="8dp"
 app:civ\_border\_width="2dp"
 app:layout\_constraintStart\_toStartOf="parent"
 app:layout\_constraintTop\_toTopOf="parent"
 app:srcCompat="@android:drawable/btn\_star\_big\_on" />

</androidx.constraintlayout.widget.ConstraintLayout>

Login

این layout برای صفحه لاگین است برای دریافت ایمیل و پسورد ودکمه لاگین است

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"

 android:layout\_width="match\_parent"
 android:layout\_height="match\_parent"
 android:background="#8decff">

 <EditText
 android:id="@+id/email\_edittext\_login"
 android:layout\_width="0dp"
 android:layout\_height="50dp"
 android:layout\_marginEnd="32dp"
 android:layout\_marginLeft="32dp"
 android:layout\_marginRight="32dp"
 android:layout\_marginStart="32dp"
 android:layout\_marginTop="32dp"
 android:background="@drawable/rounded\_edittext\_register\_login"
 android:ems="10"
 android:textColor="@android:color/black"
 android:hint="Email"
 android:textColorHint="@android:color/black"
 android:inputType="textEmailAddress"
 android:paddingLeft="16dp"
 app:layout\_constraintEnd\_toEndOf="parent"
 app:layout\_constraintStart\_toStartOf="parent"
 app:layout\_constraintTop\_toTopOf="parent" />

 <EditText
 android:id="@+id/password\_edittext\_login"
 android:layout\_width="0dp"
 android:layout\_height="50dp"
 android:layout\_marginTop="8dp"
 android:background="@drawable/rounded\_edittext\_register\_login"
 android:ems="10"
 android:paddingLeft="16dp"
 android:hint="Password"
 android:textColorHint="@android:color/black"
 android:textColor="@android:color/black"
 android:inputType="textPassword"
 app:layout\_constraintEnd\_toEndOf="@+id/email\_edittext\_login"
 app:layout\_constraintStart\_toStartOf="@+id/email\_edittext\_login"
 app:layout\_constraintTop\_toBottomOf="@+id/email\_edittext\_login" />

 <Button
 android:id="@+id/login\_button\_login"
 android:layout\_width="0dp"
 android:layout\_height="50dp"
 android:layout\_marginTop="8dp"
 android:text="Login"
 android:background="@drawable/rounded\_button"
 android:textColor="@android:color/white"
 android:textStyle="bold"
 app:layout\_constraintEnd\_toEndOf="@+id/password\_edittext\_login"
 app:layout\_constraintStart\_toStartOf="@+id/password\_edittext\_login"
 app:layout\_constraintTop\_toBottomOf="@+id/password\_edittext\_login" />

 <TextView
 android:id="@+id/back\_to\_register\_textview"
 android:layout\_width="wrap\_content"
 android:layout\_height="44dp"
 android:layout\_marginEnd="8dp"
 android:layout\_marginLeft="8dp"
 android:layout\_marginRight="8dp"
 android:layout\_marginStart="8dp"
 android:layout\_marginTop="8dp"
 android:text="Back to registration"
 android:textColor="@android:color/white"
 android:textSize="16sp"
 android:textStyle="bold"
 app:layout\_constraintEnd\_toEndOf="@+id/login\_button\_login"
 app:layout\_constraintStart\_toStartOf="@+id/login\_button\_login"
 app:layout\_constraintTop\_toBottomOf="@+id/login\_button\_login" />
</androidx.constraintlayout.widget.ConstraintLayout>

mainActivity

این اکتیویتی برای صفحه ثبت نام است .

onActivityResult

را این تابع برای انخاب عکس از گالری گوشی است .

performRegister()

این تابع برای ثبت نام است که ایمیل و پسورد را دریافت می کند و انها را در فایر بیس زخیره میکند .

uploadImageToFirebaseStorage()

عکس هایی که انتخاب کردیم در فایربیس زخیره می کنیم .

saveUserToFirebaseDatabase

اکاربران را در دیتا بیس زخیزه می کند .

onOptionsItemSelected

onCreateOptionsMenu

برای ایحاد دستورات منو است .

onOptionsItemSelected

uploadImageToFirebaseStorage()

onActivityResultا

package com.example.mymasenger
import android.app.Activity
import android.content.Intent
import android.net.Uri
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.provider.MediaStore
import android.util.Log
import android.widget.Toast

import com.example.mymasenger.MainActivity.Companion.*TAG*import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.storage.FirebaseStorage
import kotlinx.android.synthetic.main.activity\_last\_massege.\*
import kotlinx.android.synthetic.main.activity\_main1.\*
import java.util.\*

class MainActivity : AppCompatActivity() {
 companion object {
 val TAG = "MainActivity"
 }

 override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContentView(R.layout.*activity\_main1*)

 register\_button\_register.setOnClickListener **{** performRegister()
 **}** already\_have\_account\_text\_view.setOnClickListener **{** Log.d(TAG, "Try to show login activity")

 // launch the login activity somehow
 val intent = Intent(this, Login::class.*java*)
 startActivity(intent)
 **}** selectphoto\_button\_register.setOnClickListener **{** Log.d(TAG, "Try to show photo selector")

 val intent = Intent(Intent.*ACTION\_PICK*)
 intent.*type* = "image/\*"
 startActivityForResult(intent, 0)
 **}** }

 var selectedPhotoUri: Uri? = null

 override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
 super.onActivityResult(requestCode, resultCode, data)

 if (requestCode == 0 && resultCode == Activity.*RESULT\_OK* && data != null) {
 // proceed and check what the selected image was....
 Log.d(TAG, "Photo was selected")

 selectedPhotoUri = data.*data* val bitmap = MediaStore.Images.Media.getBitmap(*contentResolver*, selectedPhotoUri)

 selectphoto\_imageview\_register.setImageBitmap(bitmap)

 selectphoto\_button\_register.*alpha* = 0f

// val bitmapDrawable = BitmapDrawable(bitmap)
// selectphoto\_button\_register.setBackgroundDrawable(bitmapDrawable)
 }
 }

 private fun performRegister() {
 val email = email\_edittext\_register.*text*.toString()
 val password = password\_edittext\_register.*text*.toString()

 if (email.*isEmpty*() || password.*isEmpty*()) {
 Toast.makeText(this, "Please enter text in email/pw", Toast.*LENGTH\_SHORT*).show()
 return
 }

 Log.d(TAG, "Attempting to create user with email: $email")

 // Firebase Authentication to create a user with email and password
 FirebaseAuth.getInstance().createUserWithEmailAndPassword(email, password)
 .addOnCompleteListener **{** if (!**it**.*isSuccessful*) return@addOnCompleteListener

 // else if successful
 Log.d(TAG, "Successfully created user with uid: ${**it**.*result*?.*user*?.*uid*}")

 uploadImageToFirebaseStorage()
 **}** .addOnFailureListener**{** Log.d(TAG, "Failed to create user: ${**it**.message}")
 Toast.makeText(this, "Failed to create user: ${**it**.message}", Toast.*LENGTH\_SHORT*).show()
 **}** }

 private fun uploadImageToFirebaseStorage() {
 if (selectedPhotoUri == null) return

 val filename = UUID.randomUUID().toString()
 val ref = FirebaseStorage.getInstance().getReference("/images/$filename")

 ref.putFile(selectedPhotoUri!!)
 .addOnSuccessListener **{** Log.d(TAG, "Successfully uploaded image: ${**it**.*metadata*?.*path*}")

 ref.*downloadUrl*.addOnSuccessListener **{** Log.d(TAG, "File Location: $**it**")

 saveUserToFirebaseDatabase(**it**.toString())
 **}
 }** .addOnFailureListener **{** Log.d(TAG, "Failed to upload image to storage: ${**it**.message}")
 **}** }

 private fun saveUserToFirebaseDatabase(profileImageUrl: String) {
 val uid = FirebaseAuth.getInstance().*uid* ?: ""
 val ref = FirebaseDatabase.getInstance().getReference("/users/$uid")

 val user = User(uid, username\_edittext\_register.*text*.toString(), profileImageUrl)

 ref.setValue(user)
 .addOnSuccessListener **{** Log.d(TAG, "Finally we saved the user to Firebase Database")

 val intent = Intent(this, LastMassegeActivity::class.*java*)
 intent.*flags* = Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*.or(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*)
 startActivity(intent)

 **}** .addOnFailureListener **{** Log.d(TAG, "Failed to set value to database: ${**it**.message}")
 **}** }

}

lastMassegActivity

refreshRecyclerViewMessages()

این تابع هر بار پیامی را که ارسال می شود را لیست اضافه میکند

listenForLatestMessages()

این تابع پیام های های دریافتی را در دیتابیس زخیره می کند .

fetchCurrentUser()

verifyUserIsLoggedIn()

اگر کاربر برای اولین بار برنامه را نصب کرده باشد به صفحه ثبت نام می رود در غیر این صورت در این صفحه می ماند .

verifyUserIsLoggedIn()

verifyUserIsLoggedIn()

package com.example.mymasenger
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.Menu
import android.view.MenuItem
import androidx.recyclerview.widget.DividerItemDecoration
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.database.\*
import com.xwray.groupie.GroupAdapter
import com.xwray.groupie.GroupieViewHolder
import kotlinx.android.synthetic.main.activity\_last\_massege.\*

class LastMassegeActivity : AppCompatActivity() {
 companion object {
 var currentUser: User? = null
 val TAG = "LatestMessages"
 }

 override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContentView(R.layout.*activity\_last\_massege*)

 listenForLatestMessages()

 fetchCurrentUser()

 verifyUserIsLoggedIn()

 *supportActionBar*?.setDisplayHomeAsUpEnabled(true)
 recyclerview\_latest\_messages.*adapter* = adapter
 recyclerview\_latest\_messages.addItemDecoration(DividerItemDecoration(this, DividerItemDecoration.*VERTICAL*))

 adapter.setOnItemClickListener **{** item, view **->** Log.d(TAG, "123")
 val intent = Intent(this, ChatLogActivity::class.*java*)

 // we are missing the chat partner user

 val row = item as LatestMessageRow
 intent.putExtra(NewmassegeActivity.USER\_KEY, row.chatPartnerUser)
 startActivity(intent)
 **}** }

 val latestMessagesMap = HashMap<String, ChatMessage>()

 private fun refreshRecyclerViewMessages() {
 adapter.clear()
 latestMessagesMap.values.*forEach* **{** adapter.add(LatestMessageRow(**it**))
 **}** }

 private fun listenForLatestMessages() {
 val fromId = FirebaseAuth.getInstance().*uid* val ref = FirebaseDatabase.getInstance().getReference("/latest-messages/$fromId")

 ref.addChildEventListener(object: ChildEventListener {
 override fun onChildAdded(p0: DataSnapshot, p1: String?) {
 val chatMessage = p0.getValue(ChatMessage::class.*java*) ?: return
 latestMessagesMap[p0.*key*!!] = chatMessage
 Log.d(TAG, "create latest-messages ")
 refreshRecyclerViewMessages()
 }

 override fun onChildChanged(p0: DataSnapshot, p1: String?) {
 val chatMessage = p0.getValue(ChatMessage::class.*java*) ?: return
 latestMessagesMap[p0.*key*!!] = chatMessage
 refreshRecyclerViewMessages()
 Log.d(TAG, "create latest-messages ")
 }

 override fun onChildMoved(p0: DataSnapshot, p1: String?) {

 }
 override fun onChildRemoved(p0: DataSnapshot) {

 }
 override fun onCancelled(p0: DatabaseError) {

 }
 })
 }

 val adapter = GroupAdapter<GroupieViewHolder>()

 private fun fetchCurrentUser() {
 val uid = FirebaseAuth.getInstance().*uid* val ref = FirebaseDatabase.getInstance().getReference("/users/$uid")
 ref.addListenerForSingleValueEvent(object: ValueEventListener {

 override fun onDataChange(p0: DataSnapshot) {
 currentUser = p0.getValue(User::class.*java*)
 Log.d("LatestMessages", "Current user ${currentUser?.profileImageUrl}")
 }

 override fun onCancelled(p0: DatabaseError) {

 }
 })
 }

 private fun verifyUserIsLoggedIn() {
 val uid = FirebaseAuth.getInstance().*uid* if (uid == null) {
 val intent = Intent(this, MainActivity::class.*java*)
 intent.*flags* = Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*.or(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*)
 startActivity(intent)
 }
 }

 override fun onOptionsItemSelected(item: MenuItem?): Boolean {
 when (item?.*itemId*) {
 R.id.*menu\_new\_massege* -> {
 val intent = Intent(this, NewmassegeActivity::class.*java*)
 startActivity(intent)
 }
 R.id.*menu\_sign\_out* -> {
 FirebaseAuth.getInstance().signOut()
 val intent = Intent(this, MainActivity::class.*java*)
 intent.*flags* = Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*.or(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*)
 startActivity(intent)
 }
 }

 return super.onOptionsItemSelected(item)
 }

 override fun onCreateOptionsMenu(menu: Menu?): Boolean {
 *menuInflater*.inflate(R.menu.*nav\_menu*, menu)
 return super.onCreateOptionsMenu(menu)
 }

}

ChatMassegeRow

این کلاس هر یک از ایتم های لیست newmassege را با نام کاربر ها و پروفایل های انها که در دیتاست ذخیره شده اند لود میکند

package com.example.mymasenger
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.database.ValueEventListener
import com.squareup.picasso.Picasso
import com.xwray.groupie.GroupieViewHolder
import com.xwray.groupie.Item
import kotlinx.android.synthetic.main.lasted\_massege\_row.view.\*

class LatestMessageRow(val chatMessage: ChatMessage): Item<GroupieViewHolder>() {
 var chatPartnerUser: User? = null

 override fun bind(viewHolder: GroupieViewHolder, position: Int) {
 viewHolder.itemView.message\_textview\_latest\_message.*text* = chatMessage.text

 val chatPartnerId: String
 if (chatMessage.fromId == FirebaseAuth.getInstance().*uid*) {
 chatPartnerId = chatMessage.toId
 } else {
 chatPartnerId = chatMessage.fromId
 }

 val ref = FirebaseDatabase.getInstance().getReference("/users/$chatPartnerId")
 ref.addListenerForSingleValueEvent(object: ValueEventListener {
 override fun onDataChange(p0: DataSnapshot) {
 chatPartnerUser = p0.getValue(User::class.*java*)
 viewHolder.itemView.username\_textview\_latest\_message.*text* = chatPartnerUser?.username

 val targetImageView = viewHolder.itemView.imageview\_latest\_message
 Picasso.get().load(chatPartnerUser?.profileImageUrl).into(targetImageView)
 }

 override fun onCancelled(p0: DatabaseError) {

 }
 })
 }

 override fun getLayout(): Int {
 return R.layout.*lasted\_massege\_row* }
}

chatMessag

از کلاس زیر برای لود کردن دیتا ها در در کلاس فوق استفاده می شود .

package com.example.mymasenger

class ChatMessage(val id: String, val text: String, val fromId: String, val toId: String, val timestamp: Long) {

 constructor(): this("", "", "", "", -1)

}

کلاس user

package com.example.mymasenger

import android.os.Parcelable
import kotlinx.android.parcel.Parcelize

@Parcelize

class User (val uid: String, val username: String , val profileImageUrl: String) : Parcelable {
 constructor(): this("", "" , "")
}

chatlogactivity

listenForMessages()

پیام های ارسالی را و دریافتی را لیست می کند و انها را نشان می دهد

performSendMessage()

هر یک از پیام های ارسالی را در دیتاست firebase زخیره میکند .

listenForMessages()

package com.example.mymasenger
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.database.ChildEventListener
import com.google.firebase.database.DataSnapshot
import com.google.firebase.database.DatabaseError
import com.google.firebase.database.FirebaseDatabase
import com.xwray.groupie.GroupAdapter
import com.xwray.groupie.GroupieViewHolder
import kotlinx.android.synthetic.main.activity\_chat\_log.\*

class ChatLogActivity : AppCompatActivity() {

 companion object {
 val TAG = "ChatLog"
 }

 val adapter = GroupAdapter<GroupieViewHolder>()

 var toUser: User? = null

 override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContentView(R.layout.*activity\_chat\_log*)

 recyclerview\_chat\_log.*adapter* = adapter

 toUser = *intent*.getParcelableExtra<User>(NewmassegeActivity.USER\_KEY)

 *supportActionBar*?.*title* = toUser?.username

 listenForMessages()

 send\_button\_chat\_log.setOnClickListener **{** Log.d(TAG, "Attempt to send message....")
 performSendMessage()
 **}** }

 private fun listenForMessages() {
 val fromId = FirebaseAuth.getInstance().*uid* val toId = toUser?.uid
 val ref = FirebaseDatabase.getInstance().getReference("/user-messages/$fromId/$toId")

 ref.addChildEventListener(object: ChildEventListener {

 override fun onChildAdded(p0: DataSnapshot, p1: String?) {
 val chatMessage = p0.getValue(ChatMessage::class.*java*)
 Log.d(TAG , "List Massege")
 if (chatMessage != null) {
 Log.d(TAG, chatMessage.text)
 Log.d(TAG , "List Massege")
 if (chatMessage.fromId == FirebaseAuth.getInstance().*uid*) {
 val currentUser = LastMassegeActivity.currentUser ?: return
 adapter.add(ChatFromItem(chatMessage.text, currentUser))

 } else {
 adapter.add(ChatToItem(chatMessage.text, toUser!!))
 }
 }

 recyclerview\_chat\_log.scrollToPosition(adapter.*itemCount* - 1)

 }

 override fun onCancelled(p0: DatabaseError) {

 }

 override fun onChildChanged(p0: DataSnapshot, p1: String?) {

 }

 override fun onChildMoved(p0: DataSnapshot, p1: String?) {

 }

 override fun onChildRemoved(p0: DataSnapshot) {

 }

 })

 }

 private fun performSendMessage() {
 // how do we actually send a message to firebase...
 val text = edittext\_chat\_log.*text*.toString()

 val fromId = FirebaseAuth.getInstance().*uid* val user = *intent*.getParcelableExtra<User>(NewmassegeActivity.USER\_KEY)
 val toId = user.uid

 if (fromId == null) return

// val reference = FirebaseDatabase.getInstance().getReference("/messages").push()
 val reference = FirebaseDatabase.getInstance().getReference("/user-messages/$fromId/$toId").push()

 val toReference = FirebaseDatabase.getInstance().getReference("/user-messages/$toId/$fromId").push()

 val chatMessage = ChatMessage(reference.*key*!!, text, fromId, toId, System.currentTimeMillis() / 1000)

 reference.setValue(chatMessage)
 .addOnSuccessListener **{** Log.d(TAG, "Saved our chat message: ${reference.*key*}")
 edittext\_chat\_log.*text*.clear()
 recyclerview\_chat\_log.scrollToPosition(adapter.*itemCount* - 1)
 **}** toReference.setValue(chatMessage)

 val latestMessageRef = FirebaseDatabase.getInstance().getReference("/latest-messages/$fromId/$toId")
 latestMessageRef.setValue(chatMessage)

 val latestMessageToRef = FirebaseDatabase.getInstance().getReference("/latest-messages/$toId/$fromId")
 latestMessageToRef.setValue(chatMessage)
 }
}

ChatItem

پیام های چت را ساز ماندهی می کند

package com.example.mymasenger
import com.squareup.picasso.Picasso
import com.xwray.groupie.GroupieViewHolder
import com.xwray.groupie.Item
import kotlinx.android.synthetic.main.chat\_row\_from.view.\*
import kotlinx.android.synthetic.main.chat\_row\_to.view.\*
class ChatFromItem(val text: String, val user: User): Item<GroupieViewHolder>() {
 override fun bind(viewHolder:GroupieViewHolder, position: Int) {
 viewHolder.itemView.textview\_from\_row.*text* = text
 val uri = user.profileImageUrl
 val targetImageView = viewHolder.itemView.imageview\_chat\_from\_row
 Picasso.get().load(uri).into(targetImageView)
 }

 override fun getLayout(): Int
 {
 return R.layout.*chat\_row\_from* }
}

class ChatToItem(val text: String, val user: User): Item<GroupieViewHolder>() {
 override fun bind(viewHolder: GroupieViewHolder, position: Int) {
 viewHolder.itemView.textview\_to\_row.*text* = text

 val uri = user.profileImageUrl
 val targetImageView = viewHolder.itemView.imageview\_chat\_to\_row
 Picasso.get().load(uri).into(targetImageView)
 }

 override fun getLayout(): Int {
 return R.layout.*chat\_row\_to* }
}

Login

با دریافت ایمیل و پسورد وارد اکانت کاربرر می شود

package com.example.mymasenger
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.Toast
import com.google.firebase.auth.FirebaseAuth
import kotlinx.android.synthetic.main.activity\_login.\*

class Login : AppCompatActivity() {

 override fun onCreate(savedInstanceState: Bundle?) {
 super.onCreate(savedInstanceState)
 setContentView(R.layout.*activity\_login*)

 *supportActionBar*?.*title*= "Login"

 login\_button\_login.setOnClickListener **{** performLogin()
 **}** back\_to\_register\_textview.setOnClickListener**{** finish()
 **}** }

 private fun performLogin() {
 val email = email\_edittext\_login.*text*.toString()
 val password = password\_edittext\_login.*text*.toString()

 if (email.*isEmpty*() || password.*isEmpty*()) {
 Toast.makeText(this, "Please fill out email/pw.", Toast.*LENGTH\_SHORT*).show()
 return
 }

 FirebaseAuth.getInstance().signInWithEmailAndPassword(email, password)
 .addOnCompleteListener **{** if (!**it**.*isSuccessful*) return@addOnCompleteListener

 Log.d("Login", "Successfully logged in: ${**it**.*result*?.*user*?.*uid*}")

 val intent = Intent(this, LastMassegeActivity::class.*java*)
 intent.*flags* = Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*.or(Intent.*FLAG\_ACTIVITY\_NEW\_TASK*)
 startActivity(intent)
 **}** .addOnFailureListener **{** Toast.makeText(this, "Failed to log in: ${**it**.message}", Toast.*LENGTH\_SHORT*).show()
 **}** }

}