



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی برق

پروژه کارشناسی

طراحی و ساخت دستگاه پوشیدنی هوشمند

با قابلیت تحلیل حرکات جسمانی

نگارش

سلمان عامی مطلق

استاد راهنما

دکتر محمداعظم خسروی

مرداد ۱۴۰۱

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

به نام خدا

تاریخ: مرداد ۱۴۰۱

تعهدنامه اصالت اثر



اینجانب سلمان عامی مطلق متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است. در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

سلمان عامی مطلق

امضا

سپاس‌گزاری

بر خود لازم می‌دانم که از زحمات جناب آقای دکتر محمداعظم خسروی که در مراحل انجام این پروژه به عنوان استاد راهنما و استاد مشاور در کنار بنده بودند و حمایت همه جانبه از من داشتند، تقدیر و تشکر به عمل آورم. امید است که توانسته باشم اندکی از الطافشان را جبران کنم.

از پدر و مادر عزیزم، که بیان تشکر از ایشان، از دامنهی لغات فراگرفته در زندگی‌ام خارج است، کمال تشکر را دارم و امیدوارم ذره‌ای از زحمات بی‌منتشان را جبران کرده باشم.

در پایان از جناب آقای دکتر فرید کاویانی، دکتر امیرحسن آشنایی و مهندسی سعید دیاری بابت راهنمایی‌های بی‌چشم‌داشت و دلسوزانه‌شان، تقدیر و تشکر می‌نمایم.

سلمان عامی مطلق
مرداد ۱۴۰۱

چکیده

امروزه جای تجهیزاتی از قبیل تفلن همراه و ساعت‌های هوشمند در زندگی مردم این دوره باز شده و استفاده‌ی رایجی دارند. لذا بهبود تعاملات انسان و سامانه‌های هوشمند می‌تواند برگ برنده‌ای برای این صنعت باشد. یکی از جنبه‌های این تعامل، برقراری ارتباط بین ساعت هوشمند، تفلن همراه و پایش حرکات فیزیکی است. در این پروژه ابتدا یک ساعت هوشمند به صورت کامل طراحی شده است. این طراحی شامل سخت‌افزار، نرم‌افزار و طراحی مکانیکی است. این دستگاه از بستر بلوتوث برای برقراری ارتباط با تفلن همراه استفاده می‌کند، دارای حسگر شتاب، حسگر سلامت (پالس اکسی متر)، بازو، موتور ایجاد لرزش، شارژر باتری لیتیومی، کلیدهای لمسی و صفحه‌ی نمایش است. به کمک حسگر شتاب و پردازش سیگنال آن، متغیرهای فضایی دست اندازه‌گیری شده و به کمک پیاده‌سازی دو فیلتر کالمن، اطلاعات حسگر فیلتر شدند. این تشخیص حرکت برای مواردی مثل شمارش گام و روشن شدن صفحه نمایش در صورت بالا آمدن دست استفاده شده‌اند. برای پیاده‌سازی این فیلتر از تکنیک جاگذاری مقدار نهایی ضرایب استفاده شده است که باعث کاهش چشمگیر حجم محاسبات، کاهش حافظه‌ی مورد نیاز و افزایش سرعت اجرا است. کار انجام شده‌ی پیش‌رو، نقطه‌ی پایانی برای این پروژه نیست و می‌توان ایده‌های زیادی برای پیشرفت و مسیر آینده‌ی این پروژه متصور شد. هدایت یک بازوی رباتیک بر اساس حرکت دست به کمک فناوری اینترنت اشیاء مثالی از این ایده‌ها است.

واژه‌های کلیدی:

ساعت هوشمند، فیلتر کالمن، ریزپردازنده، تفلن همراه هوشمند، دستگاه پوشیدنی

فهرست مطالب

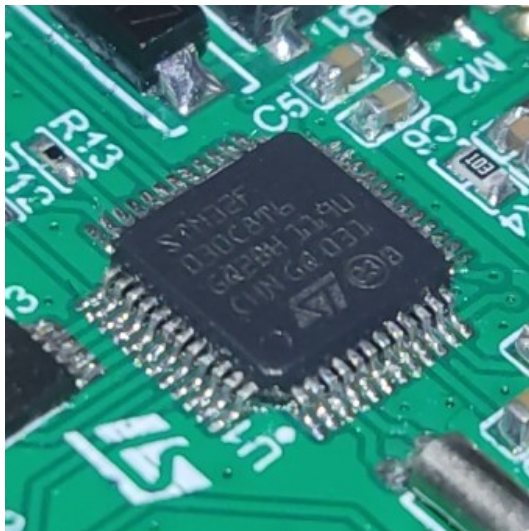
۲	۱	سخت افزار و الکترونیک
۳	۱-۱	هسته‌ی پردازشی
۹	۲-۱	درگاه بلوتوث
۱۴	۳-۱	درگاه ارتباط سریال
۱۷	۴-۱	حسگر پالس‌اکسی‌متر
۲۰	۵-۱	حسگر شتاب خطی و سرعت زاویه‌ای
۲۰	۶-۱	صفحه نمایش
۲۰	۷-۱	مدار شارژ و مدیریت توان
۲۰	۸-۱	موتور ایجاد لرزش
۲۰	۹-۱	بازر
۲۰	۱۰-۱	کلیدهای لمسی
۲۱		منابع و مراجع

شکل	فهرست اشکال	صفحه
۱-۱	تصاویری از پردازنده STM32F030	۳
۲-۱	شماتیک مربوط به بخش ریزپردازنده	۴
۳-۱	لوگوی وی ۸	۵
۴-۱	لوگوی هرِمِس و اسپایدر مانکی	۶
۵-۱	لوگوی نودجی اس	۶
۶-۱	لوگوی دِنو	۷
۷-۱	لوگوی تایپاسکرپت	۸
۸-۱	لوگوی موکا و چای	۸
۹-۱	لوگوی جِست	۹
۱۰-۱	لوگوی گِرَف کیوال	۱۰
۱۱-۱	لوگوی آپولو	۱۳
۱۲-۱	محیط Playground آپولو برای تست سرویس دهنده	۱۳
۱۳-۱	لوگوی مونگو	۱۵
۱۴-۱	لوگوی ری اکت	۱۷
۱۵-۱	آمار استفاده فریم ورک های چندسکویی جهان در سال ۲۰۱۹ [۲۵]	۱۸

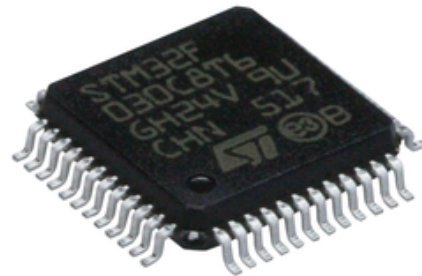
فصل اول

سخت افزار و الکترونیک

۱-۱ هسته‌ی پردازشی



(ب) مونتاژ شده روی برد پروژه



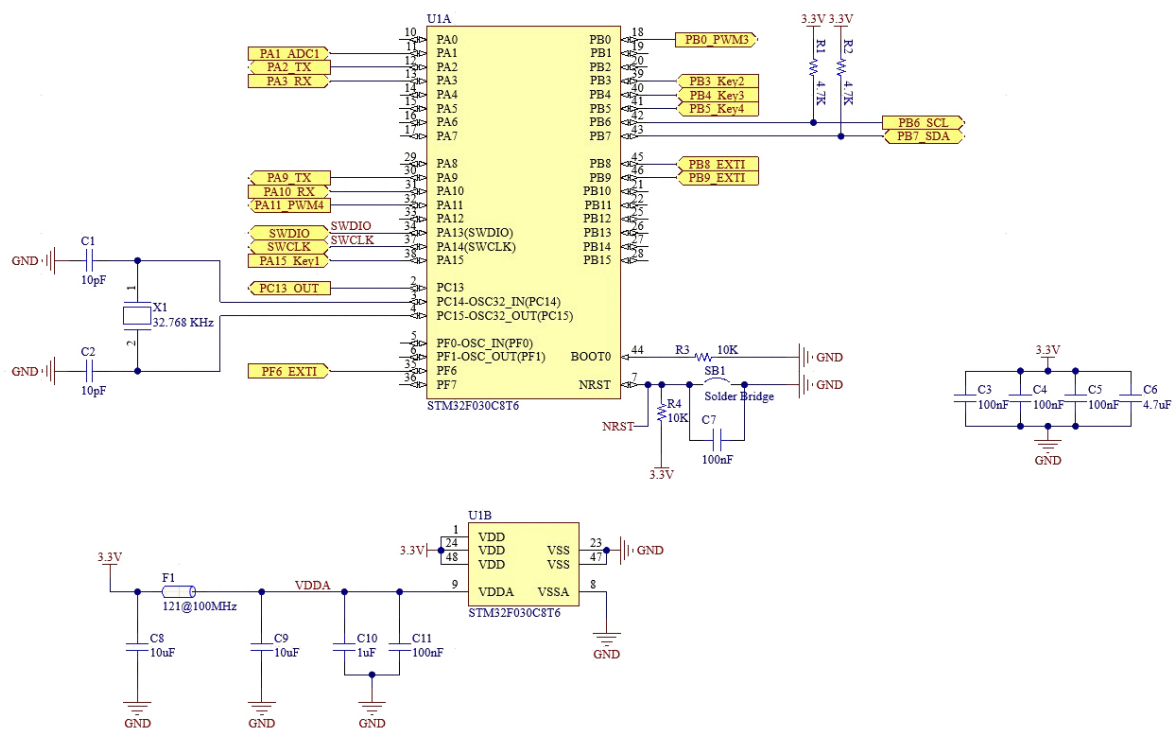
(آ) جداگانه

شکل ۱-۱: تصاویری از پردازنده STM32F030

اگر تاریخچه بوجود آمدن اکماسکریپت ۱-۱ و جاواسکریپت (شکل ۱-۲) را بررسی کنیم متوجه تقدم و تاخر زمانی عجیبی خواهیم شد. جاواسکریپت در سال ۱۹۹۶ پدید آمده، و در سال ۱۹۹۷ برای استانداردسازی به اکما ارسال شده است و پس از آن استاندارد اکماسکریپت پدید آمده! بنابراین می‌توان گفت این دو مثل مرغ و تخم مرغ بوده‌اند و با اینکه جاواسکریپت منطبق بر استاندارد اکما پیش می‌رود و نسبت به آن موخر است، اما در اولین انتشار، خودش باعث بوجود آمدن استاندارد اکما شده است! کاربرد اصلی و اولیه جاواسکریپت، ایجاد عملکرد^۱ در صفحات وب بود. درواقع، سومین تکنولوژی که توسط کنسرسیوم شبکه جهانی وب^۲ برای استفاده همگانی توسط تمامی مرورگرهای پشتیبانی‌کننده از استانداردهای وب استفاده شد، در کنار زبان‌های ایجادکننده ظاهر و قالب (شامل اچ‌تی‌ام‌ال^۳ و سی‌اس‌اس^۴) استاندارد دام^۵ بود و در عمل جاواسکریپت برای کنترل و ارائه دام توسط دو شرکت نتاسکیپ^۶ و سان میکروسیستمز^۷ طراحی شد؛ بعدها با الحاق به استاندارد اکماسکریپت و پیروی از آن، به جاواسکریپت امروزی تبدیل شد که قابلیت استفاده گسترده در مرورگرهای وب و خارج از آن‌ها را دارد [۲].

موتور جاواسکریپت گوگل وی۸ (شکل ۱-۳)، یک موتور جاواسکریپت متن‌باز^۸ است که توسط

^۱ Functionality^۲ W3C^۳ HTML^۴ CSS^۵ DOM (Document Object Model)^۶ Netscape^۷ Sun Microsystems^۸ Opensource



شکل ۱-۲: شماتیک مربوط به بخش ریزپردازنده

گوگل با زبان سی پلاس پلاس^۹ توسعه پیدا کرده و مرورگر معروف گوگل (گوگل کروم^{۱۰}) از آن استفاده می کند [۴۱].

وی ۸ با ترجمه جاواسکریپت به زبان‌های بومی و محلی ماشین هدف، باعث افزایش کارایی می‌شود. درواقع یکی از تفاوت‌های اجرای جاواسکریپت با زبانی مثل جاوا^{۱۱} در همین است. در فرآیند ترجمه زبان جاوا، خروجی تولید شده نوعی بایت‌کد^{۱۲} است که برای ماشین هدف قابل فهم نخواهد بود و ماشین مجازی جاوا^{۱۳} باید آن را تفسیر و اجرا کند. اما وی ۸ با ترجمه مستقیم جاواسکریپت به زبان بومی ماشین هدف، مرحله تفسیر و اجرای بایت‌کدهای واسط را حذف کرده و کارایی را به مقدار قابل توجهی نسبت به بقیه زبان‌های تفسیری (مانند جاوا و پایتون^{۱۴}) افزایش می‌دهد [۴۱].

وی ۸ بهینه‌سازی‌های جانبی دیگری نیز روی کد جاواسکریپت ورودی انجام می‌دهد، از جمله استفاده از نهانگاه درون‌برنامه‌ای^{۱۵}. بنابراین در کل، جاواسکریپت یکی از زبان‌هایی است که با اینکه همیشه ترجمه نمی‌شود، اما سرعت و عملکرد اجرایی آن با زبان‌های ترجمه‌شده قابل رقابت و مقایسه است [۴۱].



شکل ۱-۳: لوگوی وی ۸

همانطور که پیش‌تر گفته شد، از این موتور جاواسکریپت در چارچوب نودجی‌اس استفاده شده است. البته این موتور تنها موتور جاواسکریپت موجود نیست. برای مثال مرورگر معروف موزیلا فایرفاکس^{۱۶} از موتور جاواسکریپت اسپایدر مانکی^{۱۷} استفاده می‌کند. همچنین جلوتر در همین سند خواهیم دید که ری‌اکت نیتیو برای اجرای بهتر و سریع‌تر روی گوشی‌های هوشمند می‌تواند از موتور هرمس^{۱۸} (شکل ۱-۴) استفاده کند [۸].

^{۱۱} Java

^{۱۲} Bytecode

^{۱۳} JVM (Java Virtual Machine)

^{۱۴} Python

^{۱۵} Inline caching

^{۱۶} Mozilla Firefox

^{۱۷} SpiderMonkey

^{۱۸} Hermes



شکل ۱-۴: لوگوی هِرمِس و اسپایدر مانکی

با توجه به مطالبی که گفته شد، جاوااسکریپت در ابتدا فقط برای اجرا بر روی مرورگرهای وب بوجود آمده بود؛ اما بعدها با افزایش قدرت و توسعه کتابخانه‌ها و جامعه کاربری آن، نیاز به اجرای خارج از مرورگر وب برای جاوااسکریپت ایجاد شد. در اینجا بود که چارچوب‌هایی نظیر نودجی‌اس بوجود آمدند. درواقع یک چارچوب جاوااسکریپت، با ایجاد یک محیط اجرایی^{۱۹} مناسب در خارج از مرورگر وب، زمینه را برای اجرای کد توسط یک موتور جاوااسکریپت (مثلا وی ۸) فراهم می‌کند[۸].



شکل ۱-۵: لوگوی نودجی‌اس

در واقع درون جاوااسکریپت، اشیائی وجود دارند که به آن‌ها اشیاء میزبان^{۲۰} گفته می‌شود. برای مثال، شیء سند^{۲۱} درون کد جاوااسکریپت اجرایی در یک مرورگر تعریف شده است. اما این شیء میزبان در نودجی‌اس وجود نخواهد داشت. بنابراین جایگزین کردن اشیاء میزبان دیگر مانند اشیاء دسترسی به نظام فایل‌ها^{۲۲}، فرآیندها^{۲۳} و درخواست‌های تحت شبکه^{۲۴} با این شیء از وظایف یک چارچوب جاوااسکریپت مانند نودجی‌اس (شکل ۱-۵) است[۳۹].

چارچوب‌های معروف دیگری نیز برای جاوااسکریپت بوجود آمده‌اند، مانند چارچوب دنو^{۲۵} (شکل ۱-۶) که توسط همان سازنده اصلی نودجی‌اس بوجود آمده و درون خود از همان موتور وی ۸ استفاده می‌کند؛ با این تفاوت که اینبار بجای زبان سی‌پلاس‌پلاس، با زبان راست^{۲۶} پیاده‌سازی شده است (که

^{۱۹} Runtime environment^{۲۰} Host objects^{۲۱} document^{۲۲} Filesystem^{۲۳} Processes^{۲۴} Network requests^{۲۵} Deno^{۲۶} Rust

طبق آمار می تواند از سی پلاس پلاس سریع تر باشد). همچنین ویژگی های امنیتی بیشتری نیز به آن به نسبت نودجی اس اضافه شده است [۳۵].



شکل ۱-۶: لوگوی دِنو

برای استفاده از نودجی اس جهت اجرای سرویس دهنده ی وب، از کتابخانه های زیادی می توان استفاده کرد. معروف ترین کتابخانه موجود برای نودجی اس، اکسپرس^{۲۷} است. کتابخانه های دیگری مثل هَپی و کوآ نیز وجود دارند که به ترتیب استفاده کمتری نسبت به اکسپرس در بین جامعه کاربران نودجی اس دارند [۳۶].

زبان جاوااسکریپت از نظام نوع دهی^{۲۸} ایستا^{۲۹}ی برای متغیرها پیروی نمی کند و اکثر تایپ ها در زمان اجرا مشخص می شوند. این موضوع برای پیاده سازی پروژه های بزرگ و منسجم می تواند بسیار مشکل ساز باشد. به همین منظور، زبان های زیادی بوجود آمده اند که عملاً ویژگی های جاوااسکریپت را به ارث برده اند، اما نظام نوع دهی ایستا را نیز به آن اضافه کرده اند [۴۰].

شرکت مایکروسافت در سال ۲۰۱۲، اولین نسخه مترجم زبان تایپاسکریپت (شکل ۱-۷) را منتشر کرد. این زبان یک ابرمجموعه از تعریف زبان جاوااسکریپت است؛ به این معنا که کدهای جاوااسکریپت بدون هیچ تغییری در زبان تایپاسکریپت معتبر هستند، اما می توان برای متغیرهای موجود در آن ها نوع دهی ایستا و سخت گیرانه انجام داد. تایپاسکریپت از نوع دهی های بسیار به روز و پیشرفته ای برای مدیریت انواع حالت ها و شرایط پیچیده استفاده می کند که می تواند کیفیت و خوانایی کدهای تولید شده را بشدت افزایش داده و در عین حال، نگهداری از آن ها را بسیار ساده تر کند [۴۰].

^{۲۷} Express

^{۲۸} Type system

^{۲۹} Static



شکل ۱-۷: لوگوی تایپاسکریپت

نکته بسیار مهم در مورد تایپاسکریپت این است که خروجی تولیدشده توسط مترجم تایپاسکریپت، کد جاواسکریپت معادل آن است! بنابراین تایپاسکریپت می‌تواند بصورت یک لایه اضافه بر روی جاواسکریپت سوار شود و پس از انجام ترجمه، موتورهای جاواسکریپت ادامه کار را بدست بگیرند. یکی از مزایای این نوع ترجمه، عدم تاثیر حجم اضافه شده به کد در اثر استفاده از نظام نوع‌دهی بر روی کارایی و سرعت اجرای کد است.

تایپاسکریپت تنها راهکار بوجود آمده برای تکمیل زبان جاواسکریپت نیست؛ بلکه خروجی مترجم‌های دیگری مثل بابل^{۳۰}، دارت^{۳۱}، فلو^{۳۲}، ال‌م^{۳۳}، کافه‌اسکریپت^{۳۴} و... نیز یا بصورت مستقیم یا غیر مستقیم قابل تبدیل به جاواسکریپت هستند [۴۰].

آزمون نرم‌افزار^{۳۵} همواره یکی از پایه‌ای‌ترین مراحل تولید یک نرم‌افزار است. نرم‌افزارها و قطعه‌کدهای نوشته شده به زبان جاواسکریپت نیز از این قاعده مستثنی نیستند. در این بخش چند ابزار معروف و کاربردی برای انجام آزمون‌های مختلف را بر روی کدهای نوشته شده با زبان جاواسکریپت معرفی می‌کنیم.



شکل ۱-۸: لوگوی موکا و چای

Babel^{۳۰}Dart^{۳۱}Flow^{۳۲}Elm^{۳۳}CoffeeScript^{۳۴}Software testing^{۳۵}

موکا

موکا^{۳۶} یک چارچوب آزمون جاوااسکریپت برای نرم افزارهای تولید شده تحت چارچوب نودجی اس است، که علاوه بر اجرای بیرون از مرورگر، از قابلیت های مرورگرها نیز پشتیبانی می کند. این چارچوب همچنین قابلیت تولید گزارش های پوشش آزمون^{۳۷} رو نیز داراست [۳۷].

موکا از تمامی کتابخانه های اثبات صحت کد^{۳۸} پشتیبانی می کند. اما یکی از معروف ترین آن ها به نام چای^{۳۹} (شکل ۸-۱)، بیشتر از بقیه مورد استفاده قرار می گیرد. این کتابخانه انواع اثبات های مختلف را برای انواع عبارات و متغیرها با شیوه ای نزدیک به زبان انسان ارائه می دهد تا نوشتن آزمون ها بسیار ساده باشد.

جست

از دیگر ابزارهای تولید و اجرای آزمون های جاوااسکریپت، می توان به جست^{۴۰} (شکل ۹-۱) اشاره کرد. جست قابلیت اجرا بر روی تمامی چارچوب های موجود برای جاوااسکریپت را دارد، اما بیشتر در توسعه سمت کاربر^{۴۱} برای آزمون استفاده می شود (برای مثال در ری اکت نیتیو^{۴۲}) [۴].



شکل ۹-۱: لوگوی جست

۲-۱ درگاه بلوتوث

گرف کیوال^{۴۳} یک زبان متن باز پرس و جو^{۴۴} و ویرایش برای رابط برنامه نویسی کاربردی^{۴۵} می باشد و همچنین برای کوئری های همراه با داده نیز مناسب می باشد. گرف کیوال در ابتدا به صورت داخلی

Mocha^{۳۶}

Test coverage report^{۳۷}

Code assertion library^{۳۸}

Chai^{۳۹}

Jest^{۴۰}

Client side^{۴۱}

React Native^{۴۲}

GraphQL^{۴۳}

Query^{۴۴}

Application Programming Interface (API)^{۴۵}

توسط فیسبوک^{۴۶} در سال ۲۰۱۲ توسعه داده شد و در سال ۲۰۱۵ به صورت عمومی منتشر گردید. در تاریخ ۷ نوامبر ۲۰۱۸، پروژه گراف‌کیوال از فیسبوک به موسسه تازه تاسیس گراف‌کیوال (شکل ۱-۱۰) تحت موسسه لینوکس انتقال یافت. از سال ۲۰۱۲، رشد گراف‌کیوال با دقت بالایی، برنامه زمانی جایگزینی لی بایرون، سازنده گراف‌کیوال را دنبال کرده است. هدف بایرون این است که گراف‌کیوال را به همهی پلتفرم‌های تحت وب گسترش دهد [۳۴].



شکل ۱-۱۰: لوگوی گراف‌کیوال

کتابخانه سرورهای گراف‌کیوال برای زبان‌های متعدد و تقریباً تمام زبان‌های پرکاربرد وب موجود هستند. در فوریه ۲۰۱۸، زبان تعیین طرح (SDL) در گراف‌کیوال به بخشی از تعریف آن تبدیل شد. بعد از توسعه و استفاده داخلی برای ۳ سال، در سال ۲۰۱۶، فیسبوک مشخصات و سند مرجع پیاده‌سازی چارچوب^{۴۷} خود به نام گراف‌کیوال را منتشر کرد. این چارچوب نوع جدیدی از رابط‌های دسترسی به داده‌های مبتنی بر وب را معرفی می‌کند که به عنوان جایگزینی برای رابط‌های مبتنی بر رابط‌های برنامه‌نویسی کاربردی (مثل رست^{۴۸}) ارائه شده‌اند. این چارچوب از زمان انتشارش رشد قابل ملاحظه‌ای داشته و تعداد کاربرانش نیز در حال افزایش است.

گراف‌کیوال یک روش برای توسعه رابط‌های برنامه‌نویسی کاربردی وب^{۴۹} می‌باشد که با REST و دیگر معماری‌های وب‌سرویس دارای تفاوت‌هایی است؛ به استفاده‌کنندگان اجازه می‌دهد که ساختار داده مورد نیاز خود را تعیین کنند و ساختار داده مشابه از سرور بازگردانده خواهد شد. به این صورت از انتقال حجم زیادی از اطلاعات ناخواسته جلوگیری می‌شود؛ اما این موضوع به میزان موثر بودن عملکردهای حافظه‌نهان هم بستگی دارد. انعطاف‌پذیری و غنای زبان پرس‌وجو پیچیدگی‌هایی را نیز می‌افزاید که ممکن است برای رابط‌های برنامه‌نویسی کاربردی ساده به صرفه نباشد. این زبان شامل یک نظام نوع‌دهی^{۵۰}، زبان پرس‌وجو، معنای اجرایی^{۵۱}، تصدیق ایستا و درون‌بینی تایپ می‌باشد [۱۹].

^{۴۶} Facebook

^{۴۷} Framework

^{۴۸} REST

^{۴۹} Web

^{۵۰} Type system

^{۵۱} Semantics

از لحاظ نحوی، کوثری‌های گرف‌کیوال شبیه اشیاء جاواسکریپت یا همان جی‌سان (JSON) هستند. با این حال تفاوت آن‌ها با اشیاء معمولی جی‌سان در این است که کوثری‌ها بر اساس الگوهای نوشته می‌شوند که وب‌سرور هدف آن‌ها را پشتیبانی می‌کند. به طور غیر رسمی، این الگوها نوع اشیاء را با تعیین کردن یک مجموعه از فیلدها تعریف می‌کنند که اشیاء می‌توانند برای این فیلدها مقادیری را داشته باشند. مقادیر ممکن می‌تواند به نوع خاصی از اعداد یا اشیاء محدود شود [۱۹].

تعریف یک سرور گرف‌کیوال از دو قسمت تشکیل می‌شود؛ اول تعریفات مربوط به تایپ‌ها، و سپس تعریف توابع رافع^{۵۲} هر کدام از خصوصیات^{۵۳} تایپ‌ها. بطور پیشفرض هر خصوصیتی از تایپ‌ها که تابع رافعی برای آن تعریف نشده باشد، از یک تابع رافع بدیهی استفاده می‌کند که صرفاً مقدار آن خصوصیت را باز می‌گرداند (و تصور می‌کند آن خصوصیت صرفاً حاوی داده است و نه عملکرد اضافی). اگر نخواهیم از این توابع رافع بدیهی استفاده کنیم، باید توابع را صریحاً تعریف کنیم؛ بنابراین، معنای کوثری‌های داده شده به طور رسمی در هیچ مدل داده مشخصی تعیین نشده است. با این حال می‌توان به داده‌ای که با استفاده از رابط گرف‌کیوال به دست آمده است، به عنوان یک چشم‌انداز مجازی مبتنی بر گراف از مجموعه داده‌های اصلی نگاه کرد. این چشم‌انداز با استفاده از پیاده‌سازی تابع داخلی ذکر شده ساخته می‌شود و شکل آن مانند یک گراف متشابه با گراف مدل داده‌هاست [۱۹].

Resolver functions^{۵۲}

Fields or Properties^{۵۳}

در گرف کیوال داده‌های سرویس به شکل یک گراف هستند که توسط یک الگو نشان داده می‌شوند. در مقابل در REST اپلیکیشن‌های سرور یک لیست از endpointها را پیاده‌سازی می‌کنند. همچنین گرف کیوال یک زبان کوئری نیز تعریف می‌کند که به کاربران اجازه می‌دهد فقط آن فیلدهایی که نیاز دارند را به طور دقیق برای سرور مشخص کنند. علاوه بر این، در سرویس‌های REST کوئری‌ها به وسیله endpoint تعریف می‌شوند. هر endpoint یک مجموعه از پیش تعریف شده از فیلدها را بازمی‌گرداند که داده مربوط به یک منبع را نمایندگی می‌کند و عموماً نمی‌توان فقط بخشی از آن‌ها را درخواست کرد. از طرف دیگر، در گرف کیوال پاسخ سرور عیناً مشابه با ساختار کوئری می‌باشد و این موضوع کار با داده‌های برگردانده‌شده را راحت‌تر می‌کند [۹].

در واقع در هر ارتباط شبکه دو تاخیر اصلی وجود دارند.

۱. تاخیر ارسال^{۵۴}

۲. تاخیر انتشار^{۵۵}

استفاده از گرف کیوال بجای REST، باعث کاهش تعداد درخواست‌ها می‌شود؛ بنابراین تعداد تاخیرهای انتشار بطور قابل توجهی کاهش می‌یابد و در مقابل با توجه به افزایش حجم بدنه درخواست‌های ارسالی، تاخیر ارسال بیشتر خواهد شد. بنابراین باید بررسی کرد که در کاربردهای عادی REST کدام یک از این تاخیرها نقش بیشتری در مجموع تاخیر درخواست‌ها دارند.

اگر REST برای درخواست‌هایی با حجم بدنه سبک استفاده شود (که در APIها اکثراً چنین است)، تاخیر ارسال بسیار ناچیز خواهد بود و در مقابل زمان رفت و برگشت^{۵۶} درخواست بخش اصلی تاخیر را تشکیل خواهد داد. بنابراین در چنین درخواست‌هایی مادامی که تاخیر ارسال حجم بدنه درخواست و پاسخ آن کمتر از زمان رفت و برگشت بسته‌ها بین سرویس‌گیرنده و سرویس‌دهنده باشد، استفاده از گرف کیوال از لحاظ تاخیر در پاسخگویی به درخواست‌ها باعث کاهش تاخیر و بهبود عملکرد خواهد شد. همان‌طور که گفته شد، گرف کیوال برای بسیاری از زبان‌ها (از جمله جاوااسکریپت) دارای کتابخانه‌های متعددی جهت ایجاد سرویس‌دهنده و سرویس‌گیرنده است. این کتابخانه‌ها از لحاظ امکانات جانبی، عملکرد، پیچیدگی و بسیاری موارد دیگر قابل مقایسه با یکدیگر هستند. در این جا ما به معرفی معروف‌ترین کتابخانه گرف کیوال برای نودجی‌اس می‌پردازیم.

^{۵۴} Transmission Delay

^{۵۵} Propagation Delay

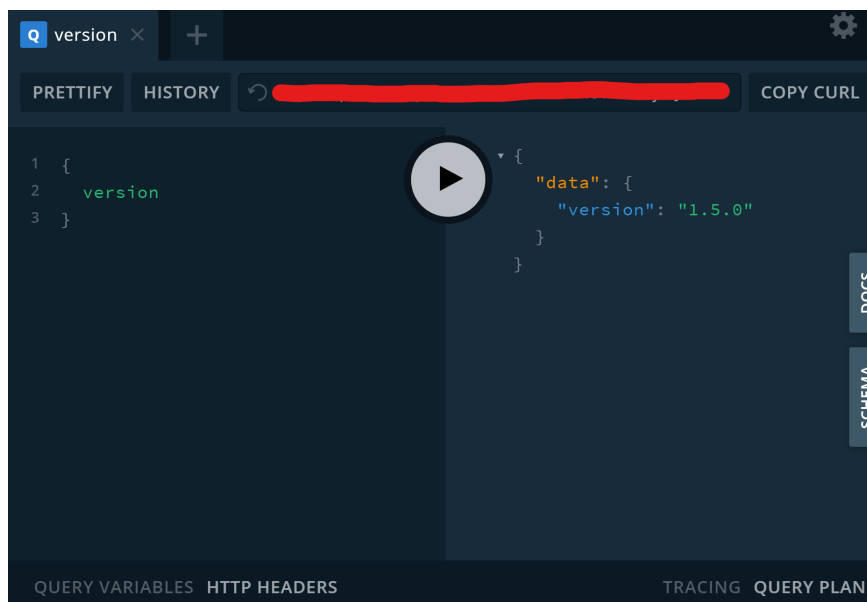
^{۵۶} Round Trip Time



شکل ۱-۱۱: لوگوی آپولو

آپولو^{۵۷}، معروف ترین و پرکاربردترین کتابخانه‌ای است که برای ایجاد سرورهای گراف کیوال در نودجی‌اس استفاده می‌شود. این کتابخانه امکانات وسیعی در زمینه استفاده از حافظه نهان در کوئری‌های گراف کیوال و همچنین ابزارهای جانبی زیادی برای ایجاد و بروزرسانی خودکار آخرین نسخه مدل داده‌ها در سرویس‌دهندگان ارائه می‌دهد. همچنین ابزارهای دیده‌بانی^{۵۸} و رهگیری^{۵۹} (شکل ۱-۱۲) کارآمدی برای بررسی عملکرد سرور ایجاد شده در اختیار مصرف‌کنندگان قرار می‌دهد [۱].

کتابخانه آپولو (شکل ۱-۱۱) در درون خودش می‌تواند از کتابخانه‌های دیگر سرویس‌دهنده وب نودجی‌اس استفاده کند. در بخش توضیحات مربوط به نودجی‌اس گفتیم که معروف ترین کتابخانه برای کاربرد گفته شده در نودجی‌اس، اکسپرس است. آپولو نیز ابزار و توابع کاملی برای سوار شدن بر روی این کتابخانه دارد و راه استفاده استاندارد از آپولو، ترکیب و سوار کردن آن بر روی اکسپرس است.



شکل ۱-۱۲: محیط Playground آپولو برای تست سرویس‌دهنده

Apollo^{۵۷}
Monitoring^{۵۸}
Tracing^{۵۹}

۳-۱ درگاه ارتباط سریال

پایگاه داده مونگو^{۶۰}، یک نرم‌افزار پایگاه‌داده^{۶۱} متمرکز ولی توزیع‌پذیر، متن‌باز^{۶۲} و رایگان است که در دسته پایگاه‌های داده NoSQL قرار می‌گیرد [۳۸]؛

در ابتدا لازم است بدانیم تفاوت این نوع پایگاه‌های داده با پایگاه‌های داده مرسوم SQL چیست. پایگاه‌های داده SQL داده‌ها را در قالب جداولی نگهداری می‌کنند که هرکدام از آن‌ها دارای تعداد محدودی ستون، و تعداد نامحدودی ردیف هستند و در هر سلول این جداول، یک مقدار، خواه عددی باشد و خواه متنی و یا حتی اطلاعات خام^{۶۳} بصورت دنباله‌ای از بایت^{۶۴}‌ها، نگهداری می‌شود. این جداول، نمی‌توانند بصورت افقی^{۶۵} رشد کنند و تنها حالت مقیاس‌پذیری آن‌ها از جهت عمودی^{۶۶}، و با اضافه شدن ردیف‌های بیشتر به آن‌هاست. برای مثال اگر نیاز باشد تا خواص یک موجودیت^{۶۷} که در این جدول‌ها با ستون‌ها نمایش داده می‌شوند را تغییر داد، تنها راه آن تغییر در طراحی کل جدول است و نمی‌توان برای ردیف‌های مختلف، خواص یا ستون‌های مختلف و یا حتی نامحدودی داشت. همچنین گاهی اوقات، ساختار داده‌های ما به گونه‌ای است که بسیاری از موجودیت‌ها، شامل موجودیت‌های دیگری هستند؛ در اینصورت در SQL‌ها باید این رابطه^{۶۸}‌ها را با تعریف کلیدهای اولیه^{۶۹} و خارجی^{۷۰} تعریف کرد و اصطلاحاً جداول را در حین اجرا با هم Join کرد. اگر این ویژگی در ساختار موجودیت‌های ما بسیار زیاد و ملموس باشد، آن‌گاه اینگونه عملیات ممکن است بازدهی و کارایی را بشدت کاهش دهد. در این شرایط، NoSQL‌ها به کمک ما می‌آیند!

بطور کلی NoSQL‌ها پایگاه‌های داده‌ای هستند که در آن‌ها ساختارهای موجودیت‌های بدون رابطه قابلیت تعریف شدن دارند و اکثراً در آن‌ها روش‌هایی برای رشد و مقیاس‌پذیری افقی داده‌ها پیش‌بینی شده است.

برای مثالی که در بخش قبلی مطرح شد، در یک پایگاه داده NoSQL مانند مونگو براحته می‌توان ساختار داده‌ای تعریف کرد که در آن مجموعه^{۷۱}‌های موجودیت‌ها از لحاظ تعداد و یا حتی نوع ویژگی‌ها با هم متفاوت هستند اما در یک مجموعه نگهداری می‌شوند؛ در صورتی که موجودیت‌های تودرتو^{۷۲} باشند،

MongoDB^{۶۰}

Database^{۶۱}

Open Source^{۶۲}

Raw^{۶۳}

Byte^{۶۴}

Horizontal Scale^{۶۵}

Vertical Scale^{۶۶}

Entity^{۶۷}

Relation^{۶۸}

Primary key^{۶۹}

Foreign Key^{۷۰}

Collection^{۷۱}

Nested^{۷۲}

نیازی به ایجاد رابطه بین مجموعه‌های مختلف برای تکمیل آن‌ها نیست و به همان حالتی که در مجموعه خود ذخیره شده‌اند، موجودیت‌هایی که صریحا زیرمجموعه آن‌ها هستند بعنوان برخی از ویژگی‌های آن موجودیت ذخیره شده است و در واقع این ویژگی‌ها مانند ستون در جدول‌های SQL هستند با این تفاوت که می‌توانند نامحدود و در طول مجموعه نابرابر و متفاوت باشند و همچنین مقادیر آن‌ها می‌تواند برخلاف SQL، هر مقداری باشد، از مقادیر عددی و رشته‌ای تا ارایه^{۷۳} و یا حتی یک مجموعه کامل دیگر!

پایگاه‌های داده NoSQL به ۴ گروه تقسیم می‌شوند:

۱. کلید-مقدار^{۷۴}: اطلاعات بصورت دوتایی کلید و مقدار ذخیره می‌شوند. مانند Redis
 ۲. ستون عریض^{۷۵}: دقیقا مشابه SQL‌ها با این تفاوت که ستون‌های جداول می‌توانند متفاوت و یا نامحدود باشند. مانند Cassandra
 ۳. سند محور^{۷۶}: اطلاعات در قالب تعدادی فایل ساختاربندی شده با مقادیری مشابه ساختار JSON نگهداری می‌شوند. مانند مونگو
 ۴. گراف محور^{۷۷}: اطلاعات در یال‌ها و گره‌های گراف نگهداری می‌شوند. مانند Neo4j
- پس از بررسی انواع NoSQL‌ها و تفاوت آن‌ها با SQL‌ها، همانطور که گفتیم، مونگو (شکل ۱-۱۳) یکی از پایگاه‌های داده از نوع سند محور است و در دسته NoSQL‌ها قرار می‌گیرد؛ با توجه به آمار اعلامی وبگاه DB-engine، مونگو محبوب‌ترین پایگاه‌داده NoSQL جهان است.



شکل ۱-۱۳: لوگوی مونگو

از ویژگی‌های منحصر بفرد مونگو، می‌توان به موارد زیر اشاره کرد^[۳۸]:

-
- Array^{۷۳}
 - Key-Value^{۷۴}
 - Wide Column^{۷۵}
 - Document based^{۷۶}
 - Graph based^{۷۷}

۱. تکثیر^{۷۸}: با استفاده از قابلیت خودکار تکثیر مونگو، می توان مجموعه‌هایی از سرورهای مونگو ایجاد کرد که هر کدام یک کپی تکراری از داده‌ها را نگهداری می کنند؛ اما تمام آن‌ها می توانند جداگانه عمل خواندن اطلاعات و پرس و جو^{۷۹} را انجام دهند (البته فقط سرور اصلی می تواند عمل نوشتن و تغییر در داده‌ها را انجام دهد). همین ویژگی باعث افزایش بسیار زیاد سرعت و کارایی می گردد.
۲. تعادل بار^{۸۰} و خوشه‌های خرد شده^{۸۱}: با استفاده از این قابلیت، داده‌های مجموعه‌های مختلف به گونه‌ای خرد می شوند که هر مجموعه سرور، مسئول نگهداری بخشی از آن‌ها باشد و هر کدام از این بخش‌ها، توسط یک کلید خاص مشخص می شوند تا بتوانند بازیابی شده و بهم متصل شوند.
۳. نگهداری و پرس و جو بهینه مختصات جغرافیایی: با استفاده از قابلیت فهرست سازی^{۸۲} از مختصات جغرافیایی و دستورات ویژه برای پرس و جوی آن‌ها که در نتیجه آن، استفاده از مونگو برای بسیاری از سیستم‌های مرتبط با مختصات جغرافیایی و یا سیستم‌های آب و هوایی بسیار بصره و بهینه شده است. همچنین مونگو قابلیت ایجاد فهرست از این مختصات‌ها را نیز داراست و این قابلیت سبب افزایش قابل توجه کارایی و سرعت اجرای پرس و جوها برای یافتن نزدیک ترین مختصات به یک مختصات خاص، از درون مجموعه‌ی بسیار بزرگی از مختصات‌ها میشود، که در بسیاری از سیستم‌ها کاربرد فراوانی دارد.
۴. ذخیره فایل و تعادل بار آن: سیستمی تحت عنوان GridFS که اجازه ذخیره فایل بعنوان داده در هر جایی از مجموعه‌ها را می دهد.
۵. خط لوله تجمیع^{۸۳}: مراحل تحت عنوان مراحل خط لوله، که می توانند در یک فرآیند تجمیع، پشت سر هم اجرا شوند و خروجی‌ای از داده‌ها بسازند که در حالت عادی ممکن بود بسیار پیچیده و ساخت آن‌ها بسیار زمان بر باشد. همچنین این مراحل بطور خودکار توسط مونگو، اصلاح و بهینه می شوند؛ برای مثال بعضی از مراحل خط لوله مونگو، وقتی تحت شرایط خاصی، قبل یا بعد از یک مرحله خاص دیگر قرار می گیرند، می توانند حذف شوند و یا در مرحله دیگری ادغام شوند؛ مونگو به طور خودکار این ساده سازی را روی تمام دستورات خط لوله تجمیع انجام می دهد.
۶. پشتیبانی از ساختار رابطه‌ای: برخلاف اکثر پایگاه‌های داده NoSQL، مونگو از ساختار رابطه‌ای نیز پشتیبانی می کند و معادلی نیز برای دستور Join در SQL دارد.

Replica Set^{۷۸}Query^{۷۹}Load Balancing^{۸۰}Sharded Clusters^{۸۱}Indexing^{۸۲}Aggregation Pipeline^{۸۳}

۴-۱ حسگر پالس اکسی متر

ری اکت جی اس^{۸۴} مجموعه ای از کتابخانه های جاوا اسکریپت برای تولید وب سایت و نرم افزارهای کاربردی تحت وب^{۸۵} است که توسط فیسبوک^{۸۶} در سال ۲۰۱۳ توسعه داده شده و منتشر شده است. در واقع وقتی یک وب سایت با ری اکت (شکل ۱-۱۴) جی اس تولید می شود، علاوه بر ظاهر آن، منطق آن نیز در کنار ظاهر پیاده سازی می شود و با توجه به توسعه روزافزون نرم افزارهای کاربردی تحت وب تک صفحه ای^{۸۷}، از ری اکت جی اس و رقباي آن (مانند انگولار^{۸۸} و فلاتر گوگل^{۸۹} و ویوجی اس^{۹۰}) استقبال بسیاری شده است [۳۰].



شکل ۱-۱۴: لوگوی ری اکت

ری اکت نیتیو^{۹۱} بر پایه یکی دیگر از سرویس های محبوب فیس بوک یعنی ری اکت جی اس می باشد که در طراحی رابط کاربری استفاده می شود. ولی برخلاف ری اکت جی اس که تمرکز اصلی آن روی مرورگر است، ری اکت نیتیو برای ساخت اپلیکیشن های موبایل استفاده می شود [۳۰]. در واقع ری اکت نیتیو یک چارچوب توسعه اپلیکیشن های چند سکویی است (که جلوتر توضیح داده خواهند شد). برنامه های ری اکت نیتیو با زبان جاوا اسکریپت و JSX نوشته می شوند. سپس می توان از این کدها برای اندروید و iOS خروجی گرفت.

نرم افزارهای کاربردی متناسب با پلتفرمی که روی آن ها اجرا خواهند شد به دو دسته تقسیم می شوند:

۱. اپلیکیشن بومی یا نیتیو^{۹۲}

۲. اپلیکیشن چندسکویی یا Cross Platform

ReactJS^{۸۴}

Web application^{۸۵}

Facebook^{۸۶}

Single paged web applications^{۸۷}

Google Angular^{۸۸}

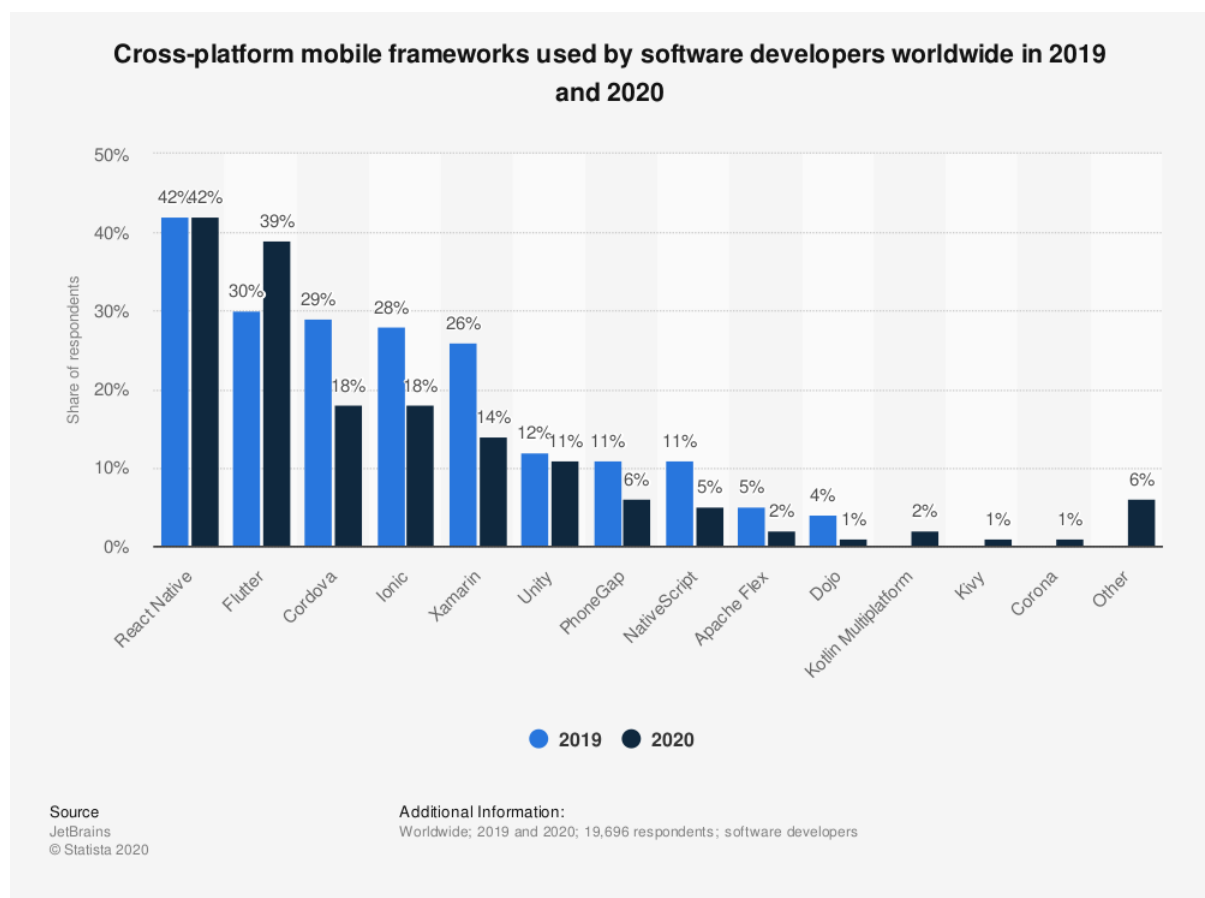
Google Flutter^{۸۹}

VueJS^{۹۰}

React Native^{۹۱}

Native^{۹۲}

به اپلیکیشن‌هایی گفته می‌شود که برای یک پلتفرم خاص توسعه داده شده‌اند. وقتی شما می‌خواهید یک اپلیکیشن نیتیو بسازید باید از زبان اصلی که برای آن پلتفرم پیشنهاد شده است استفاده کنید. برای مثال برای نوشتن یک اپلیکیشن نیتیو اندروید^{۹۳} باید از زبان کاتلین^{۹۴} (از جاوا هم برای نوشتن اپلیکیشن‌های نیتیو اندروید می‌توان استفاده کرد) استفاده کنید. وقتی از کاتلین استفاده می‌کنید اپلیکیشن شما به صورت نیتیو (بومی) فقط روی سیستم عامل اندروید اجرا خواهد شد. به اپلیکیشن‌هایی گفته می‌شود که برای چند پلتفرم توسعه داده می‌شوند. در این حالت با استفاده از یک سری فریم‌ورک‌ها می‌توان اپلیکیشنی توسعه داد که برای همه‌ی پلتفرم‌ها قابل نصب و اجرا باشد. در واقع با استفاده از یک کد قادر خواهید بود تا برای پلتفرم‌های اندروید، iOS، وب و ... خروجی قابل استفاده بگیرید. آمار استفاده از این نوع چارچوب‌ها در شکل ۱-۱۵ قابل مشاهده است.



شکل ۱-۱۵: آمار استفاده فریم‌ورک‌های چندسکویی جهان در سال ۲۰۱۹ [۲۵]

از یک دیدگاه سطح بالا به رابط کاربری برنامه‌های تلفن همراه، هر برنامه از تعدادی صفحه تشکیل می‌شود که کاربران در تعامل با عناصر موجود در رابط کاربری، بین این صفحات جابجا می‌شوند [۳۰]. رابط کاربری گرافیکی در ری‌اکت نیتیو به زبان JSX نوشته می‌شود. عناصر رابط کاربری، کامپوننت^{۹۵}

Android^{۹۳}
Kotlin^{۹۴}
Component^{۹۵}

نامیده می‌شوند. به عنوان مثال عکس‌ها، متن‌ها و دکمه‌ها، کامپوننت‌هایی هستند که به صورت از پیش تعریف شده در ری‌اکت نیتیو وجود دارند. برنامه‌نویس از کنار هم گذاشتن این عناصر و چینش آن‌ها در یک صفحه، رابط کاربری را طراحی می‌کند. همچنین کامپوننت‌های دلخواه نیز می‌تواند توسط برنامه‌نویس ایجاد گردد.

پیاده‌سازی منطق برنامه، یعنی اتصال به سرور، مدیریت داده‌ها، گرفتن ورودی‌های کاربر و پردازش آن‌ها با زبان جاوا اسکریپت پیاده‌سازی می‌گردد.

مانند هر نرم‌افزار دیگری، برنامه‌های تلفن همراه توسعه داده‌شده با ری‌اکت نیتیو، با اطلاعات کاربران سر و کار دارد. در ری‌اکت نیتیو روش‌های متعددی برای ذخیره و مدیریت داده‌ها تعبیه شده است که هر کدام از آن‌ها برحسب نیاز به کار می‌رود. روش‌ها به طور خلاصه به شرح زیر می‌باشند [۳۰]:

- متغیر^{۹۶}های معمولی جاوا اسکریپت: این نوع متغیرها برای ذخیره‌ی داده‌های میانی و موقت به کار می‌روند و نقش چرک‌نویس را دارد.
- متغیرهای ثابت^{۹۷} جاوا اسکریپت
- Props: یک کامپوننت برای نمایش‌اش ممکن است نیاز به یک سری داده‌ها داشته باشد. این داده‌ها به صورت متغیرهایی به نام props از کامپوننت بیرونی به آن‌ها منتقل می‌شوند. همانند نوع constant، این متغیرها ثابت هستند.
- State: از این نوع متغیرها برای ذخیره‌ی اطلاعات مربوط به کامپوننت استفاده می‌شود. به عنوان مثال، برای ذخیره‌ی متن نوشته شده توسط کاربرد در یک فیلد متنی، برای ذخیره‌ی وضعیت یک دکمه و... این نوع متغیرها قابل تغییر می‌باشند.
- Context: اگر بخواهیم داده‌ای را چندین لایه به کامپوننت‌های درونی‌تر پاس بدهیم، یک راه این است به کمک propsها مرحله به مرحله متغیر را به کامپوننت درونی‌تر پاس بدهیم تا سرانجام به درونی‌ترین آن‌ها برسد. اما چنین کاری، مطمئناً بهینه نمی‌باشد. راهکار بهتری که در ری‌اکت نیتیو تعبیه شده این است که چنین داده‌هایی، در یک فضای گلوبال تحت عنوان context تعریف می‌شوند و به کامپوننت‌هایی که می‌خواهیم به این داده‌های دسترسی داشته باشند، اجازه‌ی دسترسی می‌دهیم.

^{۹۶} Variable^{۹۷} Constant

۵-۱ حسگر شتاب خطی و سرعت زاویه‌ای

۶-۱ صفحه نمایش

۷-۱ مدار شارژ و مدیریت توان

۸-۱ موتور ایجاد لرزش

۹-۱ بازر

۱۰-۱ کلیدهای لمسی

منابع و مراجع

- [1] Apollo. Explore the apollo platform. <https://www.apollographql.com/docs/>, 2021. [Online; accessed 31-August-2021].
- [2] Aranda, M. What's the difference between javascript and ecma script. <https://shorturl.at/esEHN>, 2017. [Online; accessed 29-August-2021].
- [3] Atlassian. Jira software. <https://www.atlassian.com/software/jira>, 2021. [Online; accessed 20-January-2021].
- [4] Bakkhus, S. Getting started with jest. <https://jestjs.io/docs/getting-started>, 2021. [Online; accessed 30-August-2021].
- [5] Bao J., Zheng Y., Wilkie D. and M, Mokbel. Recommendations in location-based social networks: a survey. volume 19, pages 525–565. GeoInformatica, 2015.
- [6] Barone, A. Marketing strategy. <https://www.investopedia.com/terms/m/marketing-strategy.asp>, 2019. [Online; accessed 08-April-2020].
- [7] Bedgood, L. 25 stats on consumer shopping trends for 2019. <https://www.business2community.com/consumer-marketing/25-stats-on-consumer-shopping-trends-for-2019-02175812>, 2019. [Online; accessed 08-April-2021].

- [8] Biswas, N. What is the relationship between node.js and v8. <https://www.knowledgehut.com/blog/web-development/nodejs-and-v8>, 2021. [Online; accessed 29-August-2021].
- [9] Brito, Gleison and Valente, Marco Tulio. Rest vs graphql: A controlled experiment. In 2020 IEEE International Conference on Software Architecture (ICSA), pages 81–91. IEEE, 2020.
- [10] Chevalier, S. Global online shopping preference 2017, by category. <https://www.statista.com/statistics/692871/global-online-shopping-preference-by-category>, 2021. [Online; accessed 28-August-2021].
- [11] Cronin, M. What is the n+1 problem in graphql: A crash course on a surprisingly common problem. <https://shorturl.at/pwW34>, 2019. [Online; accessed 30-August-2021].
- [12] Curbside. Curbside. <https://curbside.com/>, 2021. [Online; accessed 01-January-2021].
- [13] Dark Sky. Dark sky. <https://darksky.net/>, 2021. [Online; accessed 01-January-2021].
- [14] Djordjevic, M. Statistics on online shopping vs in store shopping. <https://savemycent.com/statistics-on-online-shopping-vs-in-store-shopping>, 2021. [Online; accessed 27-August-2021].
- [15] Foursquare. Foursquare. <https://foursquare.com/>, 2021. [Online; accessed 15-January-2021].
- [16] Graphana Labs. K6 load testing. <https://k6.io/>, 2021. [Online; accessed 13-September-2021].

- [17] Grigorov, B. How to use the react context api with hooks for a clean code architecture. <https://shorturl.at/svDEV>, 2019. [Online; accessed 31-August-2021].
- [18] Hamraves. Darkube. <https://hamraves.com/darkube>, 2021. [Online; accessed 31-August-2021].
- [19] Hartig, Olaf and Pérez, Jorge. An initial analysis of facebook's graphql language. 2017.
- [20] Jackson, M. Best practices for managing and storing secrets. <https://blog.gitguardian.com/secrets-api-management/>, 2020. [Online; accessed 31-August-2021].
- [21] Joe. How to do a geonear aggregation on embedded documents. <https://shorturl.at/ptuKY>, 2020. [Online; accessed 31-August-2021].
- [22] JWT. Introduction to json web tokens. <https://jwt.io/introduction>, 2021. [Online; accessed 31-August-2021].
- [23] Kayere, P. Hapi vs koa vs express. <https://www.section.io/engineering-education/hapi-vs-koa-vs-express/>, 2020. [Online; accessed 29-August-2021].
- [24] Kotler, P. Principles of marketing. <https://www.splashcopywriters.com/blog/types-of-marketing>, 2018. [Online; accessed 08-April-2020].
- [25] Liu, S. Cross-platform mobile frameworks used by software developers worldwide in 2019 and 2020. <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours>, 2020. [Online; accessed 10-September-2020].
- [26] MongoDB Community. Geospatial queries. <https://docs.mongodb.com/manual/geospatial-queries/>, 2021. [Online; accessed 31-August-2021].

- [27] PCBWay. Pcbway website. <https://www.pcbway.com/>.
- [28] Pokémon GO. Pokémon go. <https://pokemongolive.com/>, 2021. [Online; accessed 01-January-2021].
- [29] Rabiei, H. Types of marketing. <https://parsvt.com/types-of-marketing/>, 2020. [Online; accessed 08-April-2021].
- [30] React Native Community. React fundamentals. <https://reactnative.dev/docs/intro-react>, 2020. [Online; accessed 10-September-2020].
- [31] Swarm. Swarm | remember everywhere. <https://swarmapp.com/>, 2021. [Online; accessed 15-January-2021].
- [32] Taromirad, M. and Ramsin, R. Cefam: Comprehensive evaluation framework for agile methodologies. pages 195–204. 32nd Annual IEEE Software Engineering Workshop, 2008.
- [33] Twin, A. Marketing. <https://www.investopedia.com/terms/m/marketing.asp>, 2020. [Online; accessed 08-April-2020].
- [34] Wikipedia contributors. GraphQL — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=GraphQL&oldid=978907376>, 2020. [Online; accessed 18-September-2020].
- [35] Wikipedia contributors. Deno (software) — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Deno_\(software\)&oldid=1036843366](https://en.wikipedia.org/w/index.php?title=Deno_(software)&oldid=1036843366), 2021. [Online; accessed 30-August-2021].
- [36] Wikipedia contributors. Express.js — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Express.js&oldid=1036148433>, 2021. [Online; accessed 30-August-2021].

-
- [37] Wikipedia contributors. Mocha (javascript framework) — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/Mocha_\(JavaScript_framework\)](https://en.wikipedia.org/wiki/Mocha_(JavaScript_framework)), 2021. [Online; accessed 30-August-2021].
- [38] Wikipedia contributors. Mongoddb — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=MongoDB&oldid=1042620312>, 2021. [Online; accessed 31-August-2021].
- [39] Wikipedia contributors. Node.js — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Node.js&oldid=1043654112>, 2021. [Online; accessed 28-August-2021].
- [40] Wikipedia contributors. Typescript — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=TypeScript&oldid=1042091374>, 2021. [Online; accessed 30-August-2021].
- [41] Wikipedia contributors. V8 (javascript engine) — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=V8_\(JavaScript_engine\)&oldid=1041630272](https://en.wikipedia.org/w/index.php?title=V8_(JavaScript_engine)&oldid=1041630272), 2021. [Online; accessed 30-August-2021].