



دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

گزارش کارآموزی

نام و نام خانوادگی کارآموز: مریم سعیدمهر

شماره دانشجویی: ۹۶۲۹۳۷۳

استاد کارآموزی: دکتر ناصر قدیری

سرپرست کارآموزی: مهندس احسان رفیعی

محل کارآموزی: شرکت زمین مجازی ایرانیان آینده

آدرس: اصفهان، خیابان قائم مقام فراهانی، کوچه بهمن جنوبی، پلاک ۱۱

تلفن: ۰۳۱-۳۶۶۲۹۰۸۹

تاریخ پایان: ۱۴۰۰/۰۵/۰۶

تاریخ شروع: ۱۴۰۰/۰۳/۲۲

فرم شماره 1

گزارش خلاصه ای از فعالیتهای هفتگی

شماره دانشجویی: ۹۶۲۹۳۷۳

نام و نام خانوادگی کارآموز: مریم سعیدمهر

دانشکده: برق و کامپیوتر

رشته تحصیلی: مهندسی کامپیوتر - نرم افزار

<p>راه اندازی سیستم شخصی و نصب نرم افزارها و پکیج های مورد نیاز و آموزش فشرده HTML , CSS و آشنایی با Jira (سیستم مدیریت پروژه) و Git</p>	<p>هفته اول از تاریخ: ۱۴۰۰/۰۳/۲۲ لغایت تاریخ: ۱۴۰۰/۰۳/۲۶</p>
<p>آموزش جاوا اسکریپت و مقدمه ای بر فریمورک ریئکت و انجام پروژه های تمرینی کوچک</p>	<p>هفته دوم از تاریخ: ۱۴۰۰/۰۳/۲۹ لغایت تاریخ: ۱۴۰۰/۰۴/۰۲</p>
<p>آموزش فریمورک ریئکت و انجام پروژه های تمرینی کوچک</p>	<p>هفته سوم از تاریخ: ۱۴۰۰/۰۴/۰۵ لغایت تاریخ: ۱۴۰۰/۰۴/۰۹</p>
<p>آشنایی با کتابخانه mui و انجام پروژه تمرینی to-do list همراه با اتصال به سرور تهیه شده توسط کارآموزان بک اند</p>	<p>هفته چهارم از تاریخ: ۱۴۰۰/۰۴/۱۲ لغایت تاریخ: ۱۴۰۰/۰۴/۱۶</p>

امضاء سرپرست



نام و نام خانوادگی سرپرست کارآموزی

احسان رفیعی

فرم شماره 1 (ادامه)

گزارش خلاصه ای از فعالیتهای هفتگی

شماره دانشجویی: ۹۶۲۹۳۷۳

نام و نام خانوادگی کارآموز: مریم سعیدمهر

دانشکده: برق و کامپیوتر

رشته تحصیلی: مهندسی کامپیوتر - نرم افزار

<p style="text-align: center;">هفته پنجم</p> <p>از تاریخ: ۱۴۰۰/۰۴/۱۹</p> <p>لغایت تاریخ: ۱۴۰۰/۰۴/۲۳</p> <p>شروع نسخه mvp پروژه premium tradingview</p>	
<p style="text-align: center;">هفته ششم</p> <p>از تاریخ: ۱۴۰۰/۰۴/۲۶</p> <p>لغایت تاریخ: ۱۴۰۰/۰۴/۲۹</p> <p>ادامه کار روی نسخه mvp پروژه premium tradingview</p>	
<p style="text-align: center;">هفته هفتم</p> <p>از تاریخ: ۱۴۰۰/۰۵/۰۲</p> <p>لغایت تاریخ: ۱۴۰۰/۰۵/۰۶</p> <p>رفع باگ ها و نهایی سازی نسخه mvp پروژه premium tradingview برای عرضه و تست بازار</p>	
<p style="text-align: center;">هفته هشتم</p> <p>از تاریخ: -</p> <p>لغایت تاریخ: -</p>	

امضاء سرپرست



نام و نام خانوادگی سرپرست کارآموزی

احسان رفیعی

فرم شماره 2

گزارش سرپرست کارآموزی

نام و نام خانوادگی سرپرست کارآموزی: مهندس احسان رفیعی سمت: مدیر پروژه

نام واحد صنعتی: شرکت زمین مجازی ایرانیان آینده

نام و نام خانوادگی کارآموز: مریم سعیدمهر شماره دانشجویی: ۹۶۲۹۳۷۳

گرایش تحصیلی: مهندسی کامپیوتر - نرم افزار دانشکده: برق و کامپیوتر

ردیف	اظهار نظر سرپرست کارآموز	ضعیف	متوسط	خوب	عالی
1	رعایت نظم و ترتیب و انضباط در محیط کار				✓
2	میزان علاقه و همکاری با دیگران				✓
3	علاقه به فراگیری مطالب علمی و فنی				✓
4	پیگیری وظایف و میزان پشتکار				✓
5	ارزش پیشنهادات کارآموز در جهت بهبود کار				✓
6	کیفیت گزارشهای کارآموزی (حداقل فرمهای شماره 1)				✓
7	میزان بهره گیری از امکانات موجود جهت ارتقاء توانایی علمی و فنی				✓

تعداد روزهای غیبت: 0

تعداد روزهای مرخصی: 0

پیشنهادهای سرپرست کارآموزی جهت بهبود دوره کارآموزی: -

امضاء سرپرست کارآموزی



فرم شماره 3

نظرات و پیشنهادات

(در پایان دوره تکمیل شود)

شماره دانشجویی: ۹۶۲۹۳۷۳

نام و نام خانوادگی کارآموز: مریم سعیدمهر

دانشکده برق و کامپیوتر

رشته تحصیلی: مهندسی کامپیوتر - نرم افزار

شرح نظرات و پیشنهادات:

1- در مورد دوره کارآموزی و مراحل مختلف آن:

دوره کارآموزی به جهت بهبود دانش و مهارت خود و به کارگیری مفاهیم و راهبردهای تدریس شده در دانشگاه در یک محیط واقعی بسیار مفید است. همچنین به واسطه این دوره، تجربیات کاری گرانمایی برای متمایز بودن از سایرین در صنعت مشابه کسب کرده ایم که بسیار ارزشمند می باشد. به علاوه کارآموزی فرصت مناسبی برای ایجاد شبکه ای متشکل از همکاران در حوزه صنعتی رشته تحصیلی و کاری مورد علاقه مان است. در آخر می توان گفت که در فضای رقابتی بازار کار امروزی، دوره کارآموزی به عنوان نقطه قوتی در رزومه کاری افراد و همچنین امتیاز مثبتی برای آنان محسوب می شود. در واقع افراد با گذراندن دوره کارآموزی، می توانند ویژگی های کاری منحصر به فرد خودشان را بهبود ببخشند و به راحتی در بازار کار مورد نظرشان از سایر هم تایان خود متمایز شوند.

2- در مورد امور پژوهشی، فنی و تولیدی محل کارآموزی:

شرکت زمین مجازی ایرانیان دارای جوی صمیمی و دوستانه می باشد. اعضای شرکت دارای سطح علمی و فنی بالایی هستند و متواضعانه به کارآموزها در طول این دوره کمک رسانی می کنند. همچنین کارآموزها روی پروژه های کاملاً واقعی در مقیاس بزرگ کار می کنند که این امری بسیار مهم و کمک کننده است تا اهداف دوره کارآموزی محقق شود.

توجه: علاوه بر ارائه فرم نظرات و پیشنهادات در پیوست گزارش تفصیلی، در صورت تمایل یک کپی از این فرم را به دفتر ارتباط با صنعت دانشکده تحویل نمایید.

امضاء کارآموز

فهرست مطالب

8	چکیده
9	فصل اول
9	معرفی زمین مجازی ایرانیان آینده
9	1-1- مقدمه
9	1-2- فلسفه
10	1-3- فرهنگ
10	1-4- پرسنل
11	1-5- تولیدات
12	فصل دوم
12	کارهای انجام شده در دوره کارآموزی
12	2-1- مقدمه
12	2-2- آموزش فشرده و کلی HTML
13	2-3- مروری بر CSS
13	2-4- آشنایی با نرم افزار مدیریت و کنترل پروژه Jira
16	2-5- آشنایی با گیت (Git)
17	2-6- آشنایی با زبان جاوا اسکریپت
18	2-7- آموزش فریمورک ریثکت جی اس
18	2-7-1- نصب پیش نیازها
20	2-7-2- کامپوننت ها
23	2-7-3- کار با داده ها و props
24	2-7-4- کار با استیت ها (states)
25	2-7-5- کار با استایل ها
26	2-7-6- حلقه ها
27	2-7-7- Promise

28 2-7-8- واکشی داده‌ها (fetch data)
29 2-7-9- مسیره‌ی و Routing
29 2-7-10- تست نرم‌افزار
32 2-8- آشنایی با کتابخانه Material UI
33 2-9- معرفی کلیتی از پروژه Premium TradingView
36 مراجع

چکیده

من ، مریم سعیدمهر ، دانشجوی ورودی ۹۶ مهندسی کامپیوتر هستم که در تابستان ۱۴۰۰ دوره کارآموزی خود را به عنوان کارآموز فرانت‌اند و به طور خاص کارآموز Reactjs گذراندم. در فصل اول این گزارش کارآموزی به معرفی شرکت زمین مجازی ایرانیان آینده ، زمینی که مظهر آفرینش و سکویی برای بارورسازی ایده‌های بزرگ است ، می‌پردازم. در ادامه ، در فصل دوم ، جزئیات فنی و علمی امور انجام شده در دوره کارآموزی که در فرم ۱ نیز اشاره شد خواهم پرداخت که از مهم ترین مطالب آن می‌توان به آموزش فشرده HTML, CSS, JS و React js اشاره کرد همچنین گریزی به آموزش جیرا و گیت به عنوان ابزار کنترل و مدیریت پروژه و همچنین اصول مهندسی نرم افزار از جمله اسکرام خواهم زد .

فصل اول

معرفی زمین مجازی ایرانیان آینده

1-1- مقدمه

زمین مجازی ایرانیان آینده [1]، در ۱۵ اردیبهشت ۱۳۹۵ با سه نفر هم بنیان‌گذار متولد شد. زمین مظهر آفرینش و سکویی برای بارورسازی ایده‌های بزرگ است. خلق فناوری‌های اجتماعی که می‌تواند اثرگذاری ویژه‌ای بر کیفیت زندگی و کار زمینیان داشته باشد از آرمان‌های زمین است.

1-2- فلسفه

زمین یک شرکت اینترنتی است و نه یک شرکت نرم‌افزاری؛ یعنی محدود به زمان و مکان خاصی نیست؛ خدمات ارائه می‌دهد و نه نرم‌افزار و از هر جای این کره خاکی می‌توان به آن متصل شد و با آن همکاری کرد و مهم‌تر از همه رشدش به اندازه رشد اینترنت است.

زمین یک شرکت جامعه‌محور است و نه یک شرکت مشتری‌محور؛ ما به جای داشتن چند مشتری محدود برای ایجاد یک جامعه بزرگ تلاش می‌کنیم تا بتوانیم با کمک قدرت خلاقیت و مشارکت جمعی آن‌ها یک اثر اجتماعی ایجاد کنیم.

زمین بر روی لبه‌ها و روندهای نوظهور مستقر است نه در ناحیه امن؛ جدیدترین فناوری‌های موجود در جهان

توسط توسعه دهندگان جنگجو و چالش گر زمین به کار گرفته می شود. هر روز زمینیان با دنیایی از چالش عجیب و جذاب عجین است.

زمین دارای فرهنگی مسطح است و نه سلسله مراتبی؛ همکاران زمین در یک فرهنگ و اتمسفر مسطح در قالب تیم های کوچک و صمیمی به یکدیگر متصل می شوند. همه می توانند در توسعه محصولات زمین ایده دهند و کنار هم رشد کنند.

3-1- فرهنگ

اهالی زمین در محیط کار در یک بازی سازمانی به نام بالوگیم (Balogame) فعالیت ها و تلاش های هفتگی خودشان را رقم می زنند و نشان و سکه و مدال و کاپ می گیرند، از ساعت کار شناور بهره می برند، غذای رایگان در اختیارشان است و می توانند با تلاش و همراهی درازمدت سهامدار زمین شوند.

اهالی زمین ۶ روحیه و فرهنگ زیر را در خود نهاده اند:

1. نظم شخصی بالایی دارند و به زمانبندی ها و اهداف روزانه و هفتگی پایبندی دارند.
2. جنگندگی و خودباوری بالایی دارند و در حل چالش های خود و تیمشان خودران هستند.
3. تمرکز و دقت چند وجهی در مسئولیت های خود دارند و خودسازمانده هستند.
4. قدرت یادگیرندگی و یاددهندگی بالا دارند و یادگیری مداوم جزئی از سبک زندگیشان است.
5. خلاق و نوآورند و همواره به مسیرهای غیر عادی برای حل مسائل فکر می کنند.
6. تاب آوری و پذیرش بالایی در مقابل عقاید و سلايق متفاوت از خودشان دارند.

4-1- پرسنل

زمین بر مبنای معماری مسطح و هولونیک از ۳ هولون یا شبکه تشکیل شده که هر هولون نیز خود شامل چند

تیم می شود:

- شبکه پشتیبانی (Support)
 - شامل تیم های توسعه کسب و کار و مالی - حقوقی
- شبکه توسعه (Product)
 - شامل تیم های توسعه سرور، توسعه موبایل، توسعه وب، پردازش هوشمند
- شبکه رشد (Growth)
 - شامل تیم های بازاریابی، تولید محتوا و طراحی تجربه کاربری

5-1- تولیدات

زمین تنها یک سکو یا آبر محصول دارد که نامش بالونت [2] است. بالونت مثل لگو (LEGO) است؛ یعنی سکویی است که خودش دارای چندین ابزارک عمومی و ساده است که می‌توان با سر هم کردن آن‌ها محصولات و فرآیندهای بزرگتری را طراحی کرد. ابزارک‌هایی مثل شبکه، پروژه، فرم، دفتر مالی و.... بالونت برای زمین آبر سکوی کار و زندگی است که تلاش می‌کند زندگی و کار سازمان‌یافته و نوآورانه را برای کاربران و مجموعه‌های استفاده‌کننده خود رقم زند. به عبارت بهتر بالونت یک شبکه اجتماعی یا پیام‌رسان چندوجهی و بدیع است که برای فعالیت‌های هدفمند، جدی و بزرگ طراحی شده است.

افراد فناور، تیم‌های کوچک و استارت‌آپ‌ها، مجموعه‌های مردم‌نهاد و خیریه‌ها، مدارس و مراکز آموزشی، کسب‌وکارهای کوچک و متوسط و حتی سازمان‌ها، کمپانی‌ها و نهادهای بزرگی که می‌خواهند از روندهای آینده‌نگرانه زیر بهره‌مند شوند مخاطب بالونت خواهند بود:

- چابکی و بهره‌مندی از قدرت چرخش (Business Agility)
- کار در حرکت و فعالیت ابری (Business Mobility)
- اتصال به اتمسفر اجتماعی (Connected Organization)
- نوآوری باز و جمع‌سپاری (Open Innovation & Crowdsourcing)
- مشارکت تیم محور (Team Collaboration)
- یادگیری و هوشمندی کسب و کار (Business Intelligence)
- خودکارسازی کسب و کار (Business Automation)
- درگیرسازی همکاران و بازی‌وارسازی سازمانی (Employee Engagement & Gamification)

فصل دوم

کارهای انجام شده در دوره کارآموزی

1-2- مقدمه

در این دوره کارآموزی ، من به عنوان کارآموز فرانت اند (ریئکت جی اس) در شرکت زمین مجازی ایرانیان آینده مشغول بودم که در این فصل کلیه فعالیت هایم را مشروح خواهم کرد.

به طور کلی ، در چهار هفته اول کارآموزی مشغول یادگیری و انجام پروژه های تمرینی کوچک بودم و در سه هفته پایانی روی پروژه اصلی کار می کردم.

2-2- آموزش فشرده و کلی HTML [3]

فهرست سرفصل ها و رئوس مطالب مطرح شده در این خصوص :

- مقدمات HTML : شامل معرفی تگ های html ، head و body و اشاره به style و خصوصیات مهم موجود در آن ، همچنین معرفی تگ های span و br
- تگ های text formatting : شامل تگ های em , strong , u , i , b و
- Link building: معرفی تگ a و خصوصیات href, title, target و نحوه آدرس دهی و همچنین نحوه ایجاد bookmark با استفاده از تگ a و خاصیت name
- عکس ها در صفحات وب: معرفی تگ img و خصوصیات alt , src و title

- لیست ها : معرفی تگ ol و ul
- جدول : معرفی تگ های tr , table و td
- تگ بسیار مهم div : مفهوم و کاربرد این تگ برای قالب بندی صفحه و مفهوم position و مختصات top و left و مفاهیم float و clear
- فرم در صفحه وب : معرفی تگ های option , select , textarea , input , form و datalist

3-2- مروری بر CSS [4]

فهرست سرفصل ها و رئوس مطالب مطرح شده در این خصوص :

- بررسی اشکالات گذاشتن Style به صورت inline
- معرفی تگ style
- مفهوم انتخاب گر یا selector در CSS و بررسی چند selector از قبیل id و class
- معرفی Style به صورت external و معرفی مفهوم CSS و تگ link

4-2- آشنایی با نرم افزار مدیریت و کنترل پروژه Jira [5]

جیرا یک نرم افزار اختصاصی ردیابی موضوع است که برای مدیریت پروژه ، ردیابی اشکال، ردیابی مسئله و گردش کار استفاده می شود JIRA. بر اساس سه مفهوم زیر است:

- پروژه
- مسئله
- گردش کار

JIRA توسط Atlassian Inc، یک شرکت استرالیایی توسعه یافته است. جیرا یک ابزار کراس پلتفرم است که می توان آن را با هر سیستم عاملی مورد استفاده قرار داد JIRA. دارای چند زبان است از جمله انگلیسی، فرانسوی، آلمانی، ژاپنی، اسپانیایی، و غیره JIRA.، از مای اس کیوال (MySQL)، اوراکل (Oracle)، پستگرس کیوال (PostgreSQL) و اس کیوال سرور (SQL Server) در قسمت Backend پشتیبانی می کند JIRA. را می توان با بسیاری از ابزارهای دیگر مانند Subversion، گیت (GIT)، Clearcase، Foundation Team، Concurrent Version System، Mercury، Software و غیره یکپارچه کرد.

به گفته Atlassian، جیرا برای پیگیری مسئله و مدیریت پروژه توسط بیش از 75000 مشتری در 122 کشور

جهان استفاده می شود. برخی از سازمان هایی که از جیرا در برخی مواقع برای ردیابی اشکال و مدیریت پروژه استفاده کرده اند عبارتند از Fedora Commons ، Hibernate و بنیاد نرم افزار Apache ، که از Jira و Bugzilla استفاده می کند.

مراحل ثبت نام در سایت [Jira](#) :

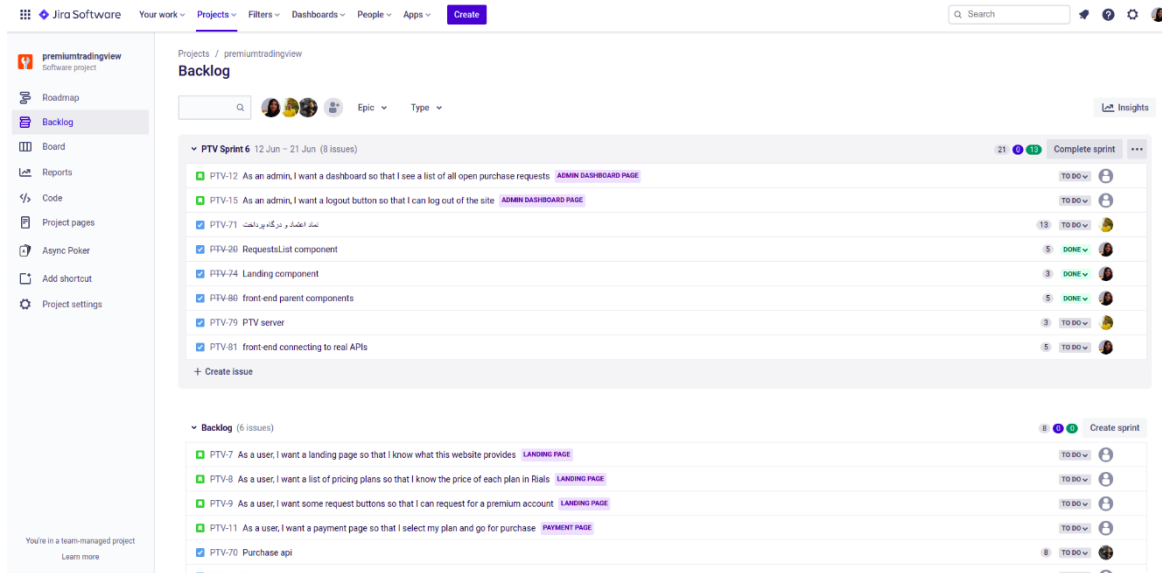
- بعد از ورود به سایت ، نسبت به ایجاد یک اکانت اقدام میکنیم.

شکل (1-2) صفحه ثبت نام در سایت جیرا

- سپس یک پروژه جدید ایجاد کرده و در صورتی که به صورت تیم روی آن کار میکنیم ، اعضای تیم را نیز به پروژه اضافه میکنیم. (اعضا نیز باید در جیرا عضو باشند).
- در نهایت بعد از ایجاد پروژه ، چندین مورد برای اضافه کردن تسک ها به صورت پیش فرض به پروژه اضافه شده اند که میتوان آن ها را شخصی ساز نیز کرد.

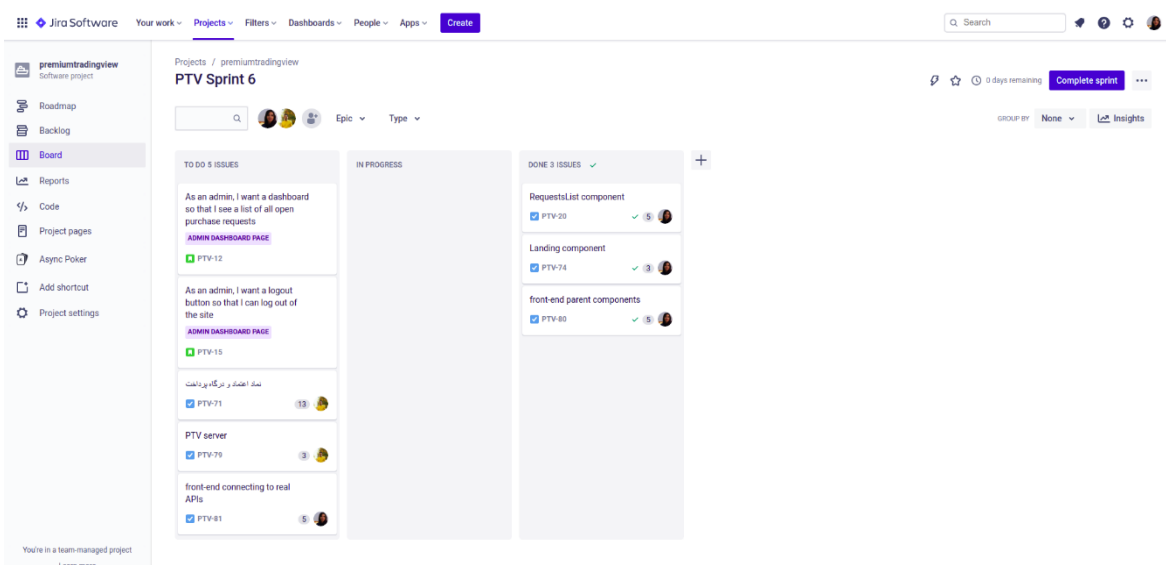
آشنایی با قسمت های مهم جیرا:

- **Backlog**: در این قسمت تسک های استخراج شده از پروژه کلی ، آورده می شوند و به اعضای تیم تخصیص داده می شوند تا در اسپرینت ها انجام شوند.



شکل (2-2) نمونه قسمت backlog سایت جیرا برای پروژه premium tradingview

- **Board**: در این قسمت ، بعد از شروع اسپرینت ها ، تسک های تخصیص داده شده نشان داده می شوند و شامل سه بورد اصلی To Do, In Progress و Done می باشد.
- بعد از شروع اسپرینت ، تمام تسک های متعلق به اسپرینت جاری در بورد To Do قرار می گیرند و هرگاه اعضا شروع به انجام تسک کنند باید آن را به بورد In Progress منتقل کنند و هنگامی که تسک به صورت کامل انجام شد (بعد از اجازه مدیر تیم) تسک به بورد Done منتقل می شود.



شکل (2-3) نمونه قسمت board سایت جیرا برای پروژه premium tradingview مربوط به اسپرینت ششم

5-2- آشنایی با گیت (Git) [6]

گیت (Git) مشهورترین و پرکاربردترین سیستم کنترل نسخه یا ورژن کنترل، در جهان است. درست است که بدون گیت می شود برنامه نویسی کرد ولی می توان با اطمینان، مدعی شد که هیچ برنامه بزرگی در جهان بدون استفاده از یک ورژن کنترل، نوشته نشده است و همچنان می شود گفت که گیت، محبوب ترین و پر استفاده ترین ورژن کنترل حال حاضر جهان است.

سرفصل های اشاره شده :

- مقدمات گیت و نصب آن (رجوع به سایت <https://git-scm.com>)
- اولین اینیت (init) و اولین کامیت : یک پوشه مربوط به پروژه ایجاد کرده و ترمینال را باز کرده و دستور `git init` را اجرا میکنیم و منتظر میمانیم تا گیت فایل های مورد نیازش برای کنترل پروژه را ایجاد کند. سپس با ایجاد هر تغییر در این پوشه ، گیت متوجه تغییرات می شود و در صورتی که تغییرات مطلوب بود میتوانیم با دستور `git add` فایل جدید را به مرحله `stage` ببریم و سپس با دستور `git commit -m "some comment"` تغییرات را به وضعیت ثبت شده ببریم.
- بررسی تاریخچه کارها و تغییرات انجام شده : با دستور `git status` میتوانیم فایل هایی که در وضعیت `stage` هستند را مشاهده کنیم و با دستور `git log` وضعیت کامیت ها و ... را مشاهده کنیم.
- آشنایی با شاخه ها یا همان برنچ ها (Branch) : در مواقعی که اشخاص متعددی روی یک پروژه کار میکنند توصیه می شود افراد برنچ های مختص خودشان را ایجاد کنند و تغییرات را در آن ایجاد کنند تا در صورت وقوع باگ ، پروژه اصلی دچار مشکل نشود ، البته بعضاً برنچ ها مربوط به فیچرهای پروژه هستند نه افراد تیم! به هر حال برنچ ها به برگشت پذیر بودن در صورت وقوع اشکال کمک شایانی میکنند. با استفاده از دستور `git checkout -b [branch name]` یک برنچ جدید ایجاد کرده و به آن سوییچ میکنیم. در ابتدا برنچ دیفالت `main` یا `master` نام دارد.
- آشنایی و استفاده از سایت گیت هاب (GitHub)
- آشنایی با `pull` , `push` : بعد از کامیت کردن تغییرات به صورت لوکال ، برای `broadcast` شدن تغییرات روی سایت گیت هاب و سیستم بقیه اعضای تیم ، باید با دستور `git push` (به شرط اتصال به اینترنت) تغییرات را `broadcast` کنیم و در صورتی که بخواهیم تغییرات `broadcast` شده را به صورت لوکال روی سیستم خود نیز دریافت کنیم باید دستور `git pull` را وارد کنیم.
- آشنایی با `pull request` : برای اینکه برنچ های فرعی روی برنچ اصلی (معمولاً به اسم `main` یا `master` است) ادغام شوند باید دستور `merge` اجرا شود ولی اجرای این دستور توسط اعضای تیم به صورت خودکامه میتواند به خراب شدن پروژه منجر شود لذا معمولاً یکی `head` تیم مسئول مرج کردن می شود و بقیه اعضا این درخواست خود را برای سرگروه با استفاده از سایت گیت هاب ارسال

می کنند و سرگروه بعد از بازبینی درخواست ها در صورتی که مشکلی وجود نداشته باشد ، برنچ مربوطه را با استفاده از دستور `git merge [branch name]` با برنچ اصلی مرج می کند.

6-2- آشنایی با زبان جاوا اسکریپت [7]

فهرست سرفصل ها و رئوس مطالب مطرح شده در این قسمت:

- ساختار برنامه در جاوا اسکریپت
- معرفی شی Document
- معرفی خصوصیت InnerHTML
- رویدادها (Events) و توابع (Functions)
- معرفی انواع داده (Data Type) و عملگرها در جاوا اسکریپت و ساختارهای کنترلی : نوع داده رشته ای (String) و نوع داده عددی (Number) و نوع داده منطقی (Boolean) و نوع داده آرایه (Array) و نوع داده شی (Object) ، عملگرهای جاوا اسکریپت (حسابی، انتسابی، مقایسه ای و منطقی) و عملگر سه حالت یا شرطی (Conditional Operator) ، ساختار کنترلی if و ساختار کنترلی Switch
- تعریف توابع
- حلقه ها (Loops): حلقه for ، حلقه while ، حلقه do-while ، حلقه for-in برای پیمایش Object ها
- رویدادها (Events): رویدادهای ماوس ، رویدادهای صفحه کلید و شی Event
- مدیریت خطا و استثنا (بلوک try-catch)
- درخت مدل شی گرای سند (Document Object Model): تغییر محتوا از طریق DOM ، تغییر Attribute ها از طریق DOM ، تغییر Style از طریق DOM ، تعیین Event ها از طریق DOM و روابط بین گره ها در درخت DOM
- اشیا جاوا اسکریپت – شی رشته (String): خصوصیات (String Length, Constructor, Prototype) ، متدهای رشته مثل indexOf ، lastIndexOf ، concat ، charCode ، Replace ، Split ، Slice ، Substr و ...
- اشیا جاوا اسکریپت – شی آرایه (Array): خصوصیت Length ، متدهای Array مثل indexOf ، Unshift ، Shift ، Pop ، Push ، toString ، Sort ، Slice ، Join ، Concat ، lastIndexOf

Reverse و ...

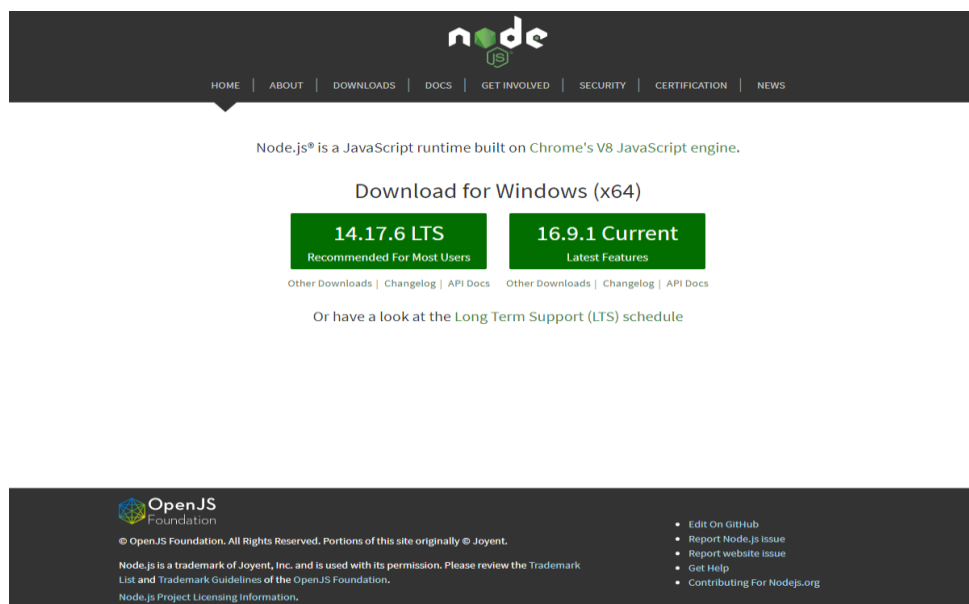
▪ اشیا جاوا اسکریپت - کلاس Math (کار با ریاضیات): خصوصیات Math ، متدهای Math مثل Floor و Round ، Round

7-2- آموزش فریمورک ریئکت جی اس [8][9]

React (ریئکت) یک کتابخانه یا فریمورک جاوا اسکریپت برای ساخت رابط های کاربری است. در حقیقت این فریمورک برای طراحی لایه View برنامه مورد استفاده قرار می گیرد.

1-7-2- نصب پیش نیازها

برای شروع به کار با هر یک از روش های معروف تولید وب اپلیکیشن ها به روش SPA ، مانند React ، Angular و Vuejs شما باید Node.js را روی سیستم خود نصب کنید. شما می توانید براحتی از سایت خود Node.js آن را دانلود کرده و روی سیستم خود نصب کنید.



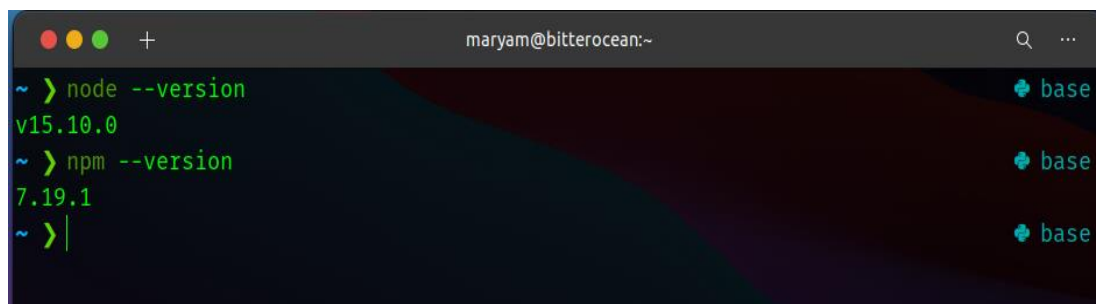
شکل (2-4) صفحه دانلود Nodejs در سایت [Nodejs.org](https://nodejs.org)

نقش Node.js در React به صورت یک سرور و بستر برای نصب و اجرای پکیجی های وابسته و مورد نیاز می باشد. با نصب Node.js یک امکان دیگر هم روی سیستم شما نصب می شود که معروف به NPM (Node Package Manager) می باشد که ما می توانیم به کمک NPM براحتی پکیج های سمت Front خود را نصب کنیم و نیاز به دانلود دستی و افزودن به پروژه را نداریم

پس NPM یا Node Package Manager یک مدیریت کننده پکیج ها می باشد که در دنیای Front بسیار کاربرد دارد و نقش مهمی دارد. البته NPM یک رقیب بسیار قوی با نام YARM دارد که بعدا شما هم از آن

استفاده خواهید کرد YARM. هم یک Package Manager بسیار قوی است که بسیار محبوب و پراستفاده می‌باشد.

پس ما Node.js را روی سیستم خود نصب کردیم. برای اینکه چک کنیم آیا Node.js از قبل روی سیستم ما نصب شده است یا نه، روش‌های مختلفی وجود دارد. یکی از روش‌ها این است که شما در ترمینال دستور – Node Version (Dash Dash – Version) را تایپ و اجرا کنید. اگر ورژنی به شما نمایش داده شد که مشخص می‌شود Node.js از قبل روی سیستم شما نصب شده است. ولی اگر پیغام خطا دریافت کردید باید Node.js را نصب کنید.

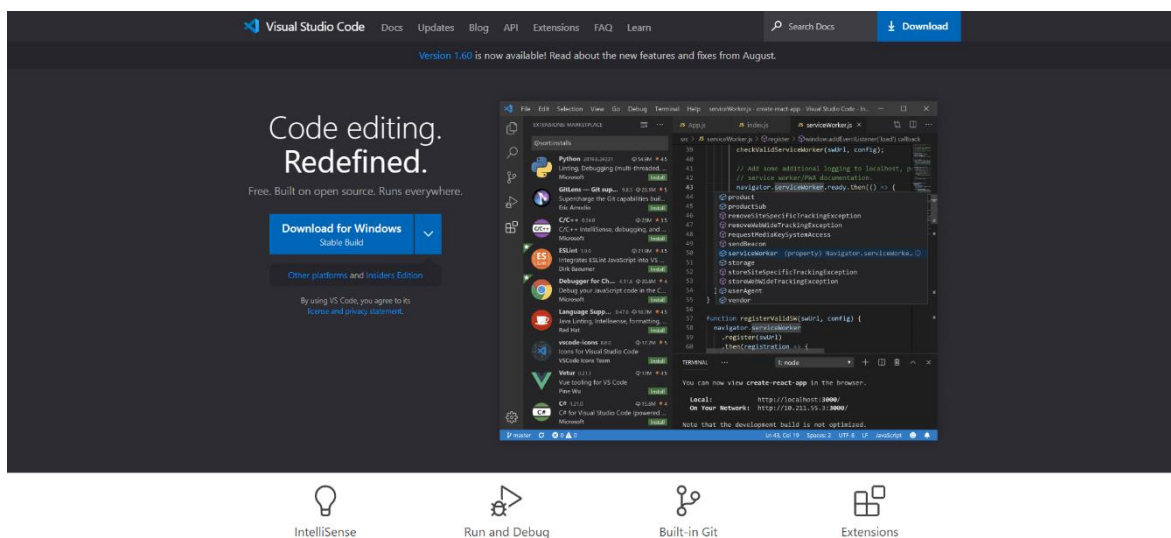


```

maryam@bitterocean:~
~ > node --version
v15.10.0
~ > npm --version
7.19.1
~ > |
  
```

شکل (2-5) مشاهده ورژن Nodejs و NPM در ترمینال

پیش نیاز اول که همان Node.js به علاوه NPM بود را آماده کردیم. حالا باید یک Code Editor خوب و سبک داشته باشیم که بتوانیم با آن برنامه خود را بنویسیم. محبوب ترین Code Editor در بین برنامه نویس‌ها در حال حاضر VS Code می‌باشد که می‌توانید از سایت خود VS Code دانلود و نصب کنید.



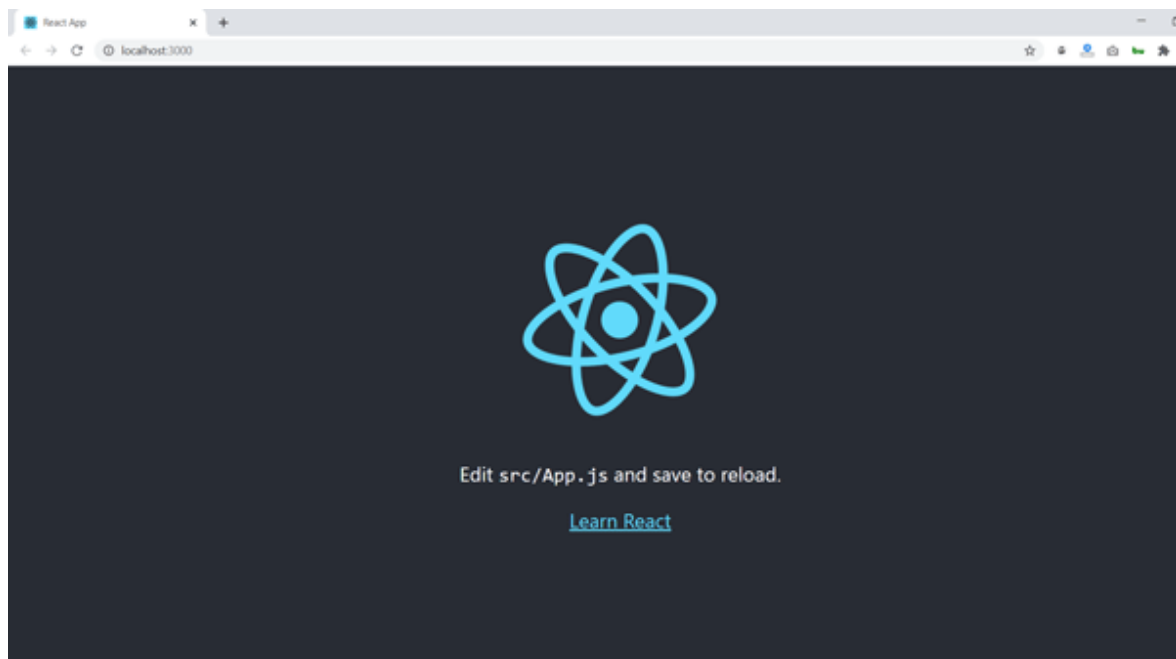
شکل (2-6) دانلود vscode از سایت رسمی آن

حال شما باید پکیج مربوط به Create-React-App را به صورت global (به صورت سراسری و عمومی) نصب کنید. برای این کار در ترمینال دستور زیر را وارد نمایید: `npm i create-react-app -g`

سوئیچ -g همان نصب به صورت global می باشد، یعنی دستورات مربوط به create-react-app در همه مسیرها قابل دسترس می باشد.

بعد از نصب پکیج بالا حالا می توانیم اولین پروژه React خود را ایجاد کنیم . توصیه میشود یک پوشه با نام مناسب ایجاد کرده و سپس در آن پوشه به کمک دستور `npm create-react-app first-app` اولین پروژه ریئکت را ایجاد کنیم.

بعد از ایجاد پروژه ، اگر دستور `npm start` را بزنیم ، این پروژه اجرا یا به اصطلاح run میشود.



شکل (2-7) خروجی دستور `npm start`

2-7-2- کامپوننت ها

کامپوننت یک ماژول مستقل است که وظیفه نمایش خروجی کدهای ری اکت در صفحه وب را به عهده دارد. ما می توانیم عناصر رابط کاربری مانند یک دکمه یا یک فیلد ورودی را به عنوان یک کامپوننت ری اکت بنویسیم. یک کامپوننت ممکن است در خروجی اش از یک یا چند کامپوننت دیگر استفاده کند. برای نوشتن برنامه های ری اکتی باید کامپوننت هایی همانند دیگر عناصر رابط کاربری بنویسیم. سپس این کامپوننت ها را داخل کامپوننت های سطح بالاتر که ساختار برنامه مان را تعریف می کرد، به طور سازمان دهی شده قرار دهیم.

برای مثال، یک فرم را در نظر بگیرید. یک فرم معمولاً شامل عناصر زیادی از قبیل فیلدهای ورودی، برچسب و دکمه ها است. هر کدام از این عناصر روی فرم می توانند به عنوان یک کامپوننت نوشته شوند. بنابراین می توانیم یک کامپوننت سطح بالاتر از خود کامپوننت فرم، ایجاد کنیم.

در کامپوننت فرم، ساختار فرم برنامه مان را تعریف کرده و هر کدام از عناصر رابط کاربری را درون آن قرار می دهیم. هر کامپوننت در یک برنامه ری اکتی از یک سری اصول مدیریت داده پیروی می کند. رابط های کاربری تعاملی پیچیده، اغلب با داده ها و وضعیت های پیچیده سر و کار دارند. ری اکت تنها محدود به کار با ظاهر یک برنامه تحت وب می باشد و هدفش این است که به ما ابزاری بدهد تا بتوانیم طرز کار برنامه را پیش بینی کنیم .

تمام کامپوننت های ری اکت حداقل به یک تابع `render()` نیاز دارند. تابع `render()` یک نمایشی از DOM مجازی از عنصرهای DOM مرورگر را به عنوان خروجی بر می گرداند.

فرمت کلی یه کامپوننت فرزند به صورت زیر است :

```
const Component_Name = () => {
  some variables, functions etc ...
  return <div></div>;
};
export default Component_Name;
```

گاهی اگر کامپوننت تنها شامل جمله `return` باشد آنگاه می توان آن را به شکل زیر باز نویسی کرد :

```
const Component_Name = () => <div></div>;
export default Component_Name;
```

اما چه زمانی کامپوننت ما در صفحه وب به کار گرفته می شود یا اصطلاحاً رندر می شود ؟ زمانی که آن کامپوننت به صورت مستقیم یا غیرمستقیم در `ReactDOM.render()` فراخوانی شود. این تابع معمولاً در به صورت دیفالت در فایل `index.js` در پوشه `src` پروژه ای که ایجاد کرده اید وجود دارد و به صورت دیفالت نیز کامپوننت `App` را فراخوانی می کند و این بدین معنی است که کامپوننت های فرزندی که می نویسیم باید نهایتاً در کامپوننت والد `App` فراخوانی شوند.

به عنوان نمونه در شکل (2-8) کامپوننت `App` نوشته شده برای نسخه `mvp` پروژه `premium tradingview` را مشاهده می کنید که کامپوننت های `Landing`, `RequestForm`, `StatusPage`, `Login` و `requestList` را رندر می کند.

```

import React from 'react';
import { Route, Switch } from 'react-router-dom';
import Header from './Header';
import Landing from './Landing';
import Login from './Login';
import RequestList from './RequestList';
import RequestForm from './RequestForm';
import StatusPage from './StatusPage';

const App = () => (
  <div>
    <Header />
    <Switch>
      <Route exact path="/" component={Landing} />
      <Route exact path="/purchase" component={RequestForm} />
      <Route
        exact
        path="/purchase/status/:status"
        component={
          routerProps => <StatusPage status={routerProps.match.params.status} />
        }
      />
      <Route exact path="/admin/login" component={Login} />
      <Route exact path="/admin/requestList" component={RequestList} />
    </Switch>
  </div>
);

export default App;

```

شکل (8-2) کامپوننت App به عنوان کامپوننت والد که کامپوننت هایی نظیر Login و Landing را فراخوانی کرده است.

در index.html یک تگ dev با id برابر با root وجود دارد که در ادامه قرار است کدهای جاوااسکریپت ما

در این قسمت رندر شوند

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="App name"
    />
    <title>App name</title>
  </head>
  <body dir="rtl">
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>

```

شکل (9-2) محتویات فایل index.html

برای رندر شدن کامپوننت مادر App در این تگ باید این کامپوننت را در تابع `ReactDOM.render()` فراخوانی کنیم. همچنین این تابع یک پارامتر دیگر نیز نیاز دارد که بیان کند این کامپوننت (کدهای جاوااسکریپت) در کدام قسمت از فایل html باید رندر شود.

```
ReactDOM.render(
  <React.StrictMode>
    <ThemeProvider theme={theme}>
      <RTL>
        <BrowserRouter>
          <App />
        </BrowserRouter>
      </RTL>
    </ThemeProvider>
  </React.StrictMode>,
  document.getElementById('root'),
);
```

شکل (2-10) فراخوانی کامپوننت App در ReactDOM

3-7-2- کار با داده‌ها و props

قرار دادن داده‌ها بطور مستقیم در داخل برنامه کار خوبی نیست. تا اینجا صرفاً کامپوننت نوشتن را یاد گرفتیم ولی این برنامه نمی‌تواند از قدرت ری‌اکت برای نمایش داده‌های داینامیک یا پویا استفاده کند. در ادامه افزودن قابلیت کار با داده‌ها در ری‌اکت را بررسی می‌کنیم.

ری‌اکت به ما اجازه می‌دهد تا به همان طریقی که به یک عنصر html داده‌ای را ارسال می‌کنیم، به کامپوننت ری‌اکت هم توسط خصیصه یا پروپرتی داده‌ای را ارسال کنیم.

اینکار مشابه خصیصه src برای یک تگ تصویر است. ما می‌توانیم به پروپرتی‌های تگی مانند `` همانند یک prop نگاه کنیم که به کامپوننت‌های ری‌اکت اعمال می‌شود.

همچنین برای دسترسی به مقادیر این پروپرتی‌ها در داخل کامپوننت از یک ورودی برای تابع نوشته شده استفاده می‌کنیم.

```
const RequestItem = (props) => {
  const {
    itemKey,
    name,
    phoneNumber,
    requestType,
    setModalOpen,
  } = props;
  return <div />;
};
export default RequestItem;
```

شکل (2-11) نمونه کد کار با prop‌ها در کامپوننت‌ها

همانطور که در شکل (11-2) آمده ، به داده هایی که به کامپوننت پاس داده می شوند در خود کامپوننت با استفاده از یک ورودی که برای آن تعریف می کنیم (در این شکل props که البته هر نام دیگری نیز میتوان استفاده کرد) دسترسی داریم.

در این شکل داده هایی تحت عنوان `itemKey`, `name`, `phoneNumber`, `requestType` و `setModalOpen` برای کامپوننت `RequestItem` فرستاده میشود.

در ادامه در شکل (12-2) نحوه فراخوانی این کامپوننت و نحوه ست کردن props های آن در کامپوننت مادر نشان داده شده است.

```
<List>
{
  requestList.map(value => (
    <div key={value.itemKey}>
      <RequestItem
        itemKey={value.itemKey}
        name={value.name}
        phoneNumber={value.phoneNumber}
        requestType={value.requestType}
        setModalOpen={setConfirmModalInfo}
      />
    </div>
  ))
}
</List>
```

شکل (12-2) نمونه کد فراخوانی کامپوننت دارای props و نحوه ست کردن این پراپ ها در کامپوننت فراخوانی کننده

شکل (12-2) قسمتی از کامپوننت `RequestList` است که به تعداد درخواست هایی که برای سیستم `tradingview` می آید قرار است کامپوننت `RequestItem` را رندر کند.

در ادامه درباره حلقه روی آرایه و انواع استیت ها توضیح خواهیم داد..

4-7-2- کار با استیت ها (states)

ری اکت به دلایلی به ما اجازه تغییر props را بر روی کامپوننت ها نمی دهد فرض کنید اگر یک پروپرتی `itemKey` به کامپوننت `RequestItem` پاس دهیم و کامپوننت `RequestItem` بتواند آن را تغییر دهد، از کجا بفهمیم پروپرتی `itemKey` در کامپوننت `RequestItem` چه مقداری دارد؟

نکته مهمی که باید بدانید این است که در کل تغییر دادن متغیری که کامپوننت والد به کامپوننت فرزند ارسال می

کند، کار خوبی نیست. با این حال، گاهی اوقات یک کامپوننت نیاز دارد که وضعیت خودش را تغییر دهد. برای مثال یک برجسب flag active تنظیم کنیم تا اگر یک کامپوننت فرزند نیاز داشت بتواند زمان را در کامپوننت ساعت انتخاب یا بروزرسانی کند.

stateها تنها در کامپوننت اصلی و کامپوننت های فرزند آن قابل دسترس هستند. همانند روشی که ما با آن یک props را به یک کامپوننت ارسال می کردیم، stateها می توانند توسط state در یک کامپوننت قابل دسترس باشند. هر زمان که یک وضعیت تغییر کرد (که اینکار توسط تابع setState انجام می شود)، کامپوننت رندر خواهد شد. به طور مثال در شکل (2-13) در کامپوننت Landing یک دکمه خرید داریم و قرار است اگر کاربر سایت روی دکمه خرید کلیک کرد، به آدرس /purchase منتقل شود که این کار را با استیت ها هندل کرده ایم به این صورت که یک استیت با عنوان shouldRedirect تعریف کرده ایم که نوع بولین دارد و در صورتی که روی دکمه ی خرید کلیک شود مقدار این استیت با استفاده از تابع setter آن به مقدار true تغییر می کند و در نتیجه تغییر این استیت منجر به ری رندر شدن این کامپوننت می شود و در صورت رندر شدن مجدد، از آنجایی که مقدار این استیت برای با true است لذا در عبارت شرطی ابتدای return دستور <Redirect /> اجرا میشود و هدف اولیه محقق می گردد.

```
const Landing = () => {
  const [shouldRedirect, setShouldRedirect] = useState(false);
  const purchaseClickHandler = () => setShouldRedirect(true);

  return shouldRedirect
    ? <Redirect to="/purchase" />
    : <Button
      onClick={purchaseClickHandler}
      fullWidth
      variant="contained"
      color="primary"
    >
      خرید
    </Button> ;
};
export default Landing;
```

شکل (2-13) نمونه کد کار با استیت ها در کامپوننت

5-7-2- کار با استایل ها

هیچ برنامه ای بدون استایل، کامل نیست. در این درس به نحوه استایل دهی کامپوننت های ری اکت توسط CSS می پردازیم.

برای استایل دهی به کامپوننت های ری اکت چندین روش وجود دارد:

1. فایل های CSS

استفاده از CSS برای استایل دهی به برنامه های تحت وب امری رایج است. در حقیقت CSS یک روش برای استایل دهی به عناصر HTML است که بصورت مجزا از کدهای HTML نوشته می شود. استفاده از CSS در ری اکت هم نیاز به کار اضافه ای ندارد و به همان شیوه معمول انجام می شود. برای اینکار باید به کامپوننت های ری اکت، یک id یا کلاس هایی را نسبت بدهیم و سپس توسط سلکتورهای CSS، کامپوننت دلخواه را انتخاب کرده و استایل مورد نظرمان را روی آن اعمال می کنیم.

2. استایل های درونی (inline styles)

اضافه کردن استایل ها به کامپوننت نه تنها ما را قادر می سازد تا بتوانیم در داخل کامپوننت ها، استایل تعریف کنیم بلکه می توانیم بصورت پویا و بسته به وضعیت (state) های مختلف در برنامه، استایل های خود را تعریف کنیم. ری اکت به جای استفاده از یک فایل جداگانه CSS، از آبجکت های جاوا اسکریپت برای تعریف استایل استفاده می کند.

```
import React from 'react';
import makeStyles from '@material-ui/core';

const useStyles = makeStyles(theme => ({
  logo: {
    maxWidth: 60,
    marginRight: theme.spacing(1),
  },
}));

const Header = () => {
  const classes = useStyles();
  return <img src={logo} alt="Logo" className={classes.logo} />;
};

export default Header;
```

شکل (2-14) نمونه کد استفاده از inline style در کامپوننت

6-7-2- حلقه ها

تابع map یکی از توابع داخلی جاوا اسکریپت است که بر روی آرایه ها استفاده می شود. این تابع یک کالбек (یا تابع) را به عنوان پارامتر گرفته و آن را بر روی تمام عناصر آرایه اجرا می کند. در شکل (2-12) نمونه استفاده از map را مشاهده می کنید.

برای افزایش کارایی، ری اکت با استفاده از DOM مجازی سعی می کند تا تعداد عناصر DOM که در هنگام نمایش ویوها نیاز است را محدود کند. اگر تغییری اتفاق نیفتد، ری اکت هیچ بروزرسانی در مرورگر انجام نمی دهد. این کار یک ویژگی خوب برای ساخت برنامه های تحت وب است، اما گاهی اوقات باید یک id منحصر به فرد برای عناصرمان اضافه کنیم. پیمایش در یک لیست و نمایش کامپوننت ها در متد map یکی از آن مواقعی است که باید از این قابلیت استفاده کنیم.

ری اکت یک پروپرتی ویژه به نام key دارد که به هر کدام از عناصر یک لیست اختصاص می یابد. مقدار پروپرتی key هر چیزی می تواند باشد، اما باید برای آن عنصر منحصر به فرد باشد.

Promise -2-7-7

طبق تعریف موزیلا، آبجکت promise برای مدیریت محاسبات غیرهمزمان در مواقعی که مدیریت محاسبات توسط متدهای کالباک مشکل باشد، استفاده می شود.

Promise ها متدهایی را برای کنترل مقادیر، در حین انجام موفقیت آمیز یا غیرموفقیت آمیز یک عملیات، ارائه می کنند

Promise با ما این امکان را می دهد تا در مواقع موفقیت یا شکست یک عملیات غیرهمزمان، یک قابلیت را به اجرا در آوریم. همچنین این اجازه را می دهد تا این گونه سناریوهای پیچیده را با کدهای معمولی (همزمانی) پیاده سازی کنیم.

یک promise در یک زمان تنها می تواند یکی از حالت های زیر را داشته باشد:

- معلق (pending)
- انجام شده (resolved)
- رد شده (error)

یک آبجکت معلق (Pending) می تواند در یکی از حالت های انجام شده (resolved) یا رد شده (error) و تنها یکبار اتفاق بیفتد و در بعضی از مواقع که ممکن است خطاهای پیچیده در برنامه اتفاق بیفتد، کاربرد دارد. در این شرایط، هر حالت که اتفاق بیفتد یک promise را بر می گردانیم. اگر بخواهیم مقداری را از تابعی که از promise استفاده می کند را برگردانیم، باید promise جدیدی را ایجاد کنیم.

در ادامه درباره fetch و axios صحبت می‌کنم که در استفاده از آن‌ها از promise استفاده زیادی می‌شود

8-7-2- واکنشی داده‌ها (fetch data)

در شکل (15-2) این تابع ری‌اکت درخواست لاگین یک کاربر را به سرور ارسال کرده و پاسخ سرور مبنی بر مجاز بودن یا نبودن لاگین را دریافت و به ما بر می‌گرداند.

ابتدا کامپوننت صفحه Login را ایجاد کرده و بعد از دادن اطلاعات مورد نیاز برای لاگین توسط کاربر و فشردن دکمه ورود، درخواست را به سرور ارسال و پاسخ آن را مانند زیر واکنشی می‌کنیم (پاسخ سرور در صورت مجاز بودن لاگین، شامل یک کد وضعیت ۲۰۰ و یک توکن است و در صورت غیرمجاز بودن تنها شامل کد وضعیت متناسب با دلیل غیرمجاز بودن لاگین است).

```
import axios from 'axios';
import { API_LOGIN } from '../apiConstants';

const login = async (password, setShouldRedirect) => {
  await axios.post(`${API_LOGIN}`, password)
    .then((res) => {
      if (res.status === 200) {
        localStorage.setItem('token', res.data[0].token);
        setShouldRedirect(true);
      }
    });
};

export default login;
```

شکل (15-2) نمونه کد واکنشی داده از سرور

همان‌طور که از شکل (15-2) پیداست، من از متد axios برای واکنشی داده‌ها استفاده کرده‌ام. این متد از promise‌ها پشتیبانی می‌کند. هنگامی که متد axios را فراخوانی کنید، یک promise را به ما بر می‌گرداند. به این ترتیب می‌توانیم درخواست‌ها را مدیریت کنیم. در این قسمت قصد داریم یک درخواست به سرور API مان ارسال کنیم. حال درخواست مان را در قالب یک URL به سرور ارسال کرده تا سرور دستور اجازه ورود یا منع ورود را به ما برگرداند.

اگر پاسخ سرور در مقابل درخواست واکنشی داده ما شامل مجموعه از داده‌ها باشد که باید نمایش داده شوند، کافیت با استفاده از حلقه‌ها روی دیتای واکنشی شده، فیلترهایی را اعمال کرده و دیتا را به شکل قابل عرضه تبدیل کنیم.

9-7-2- مسيردهی و Routing

تا به اینجاى آموزش، برنامه ما فقط محدود به یک صفحه بود. اما کمتر پیش می آید که یک برنامه پیچیده تنها از یک صفحه یا ویو تشکیل شده باشد. برای نمونه، یک برنامه ممکن است شامل یک صفحه لاگین باشد تا کاربران از طریق این صفحه به سایت لاگین کنند، یا یک صفحه جستجو که کاربران در آن یک عبارت را جستجو کنند. اینها ویوهای مختلف با ساختارهای متفاوتی هستند. حال می خواهیم ببینیم چطور می شود این ساختار را در برنامه خود پیاده کنیم.

در این برنامه برای مدیریت لینک صفحات مختلف از کتابخانه react-router استفاده می کنیم. بنابراین قبل از استفاده از این کتابخانه، ابتدا باید آنرا با دستور زیر نصب کنیم:

```
npm install --save react-router-dom
```

بعد از نصب react-router چند پکیج را از داخل این کتابخانه وارد (import) برنامه کرده و ساختار برنامه را بروزرسانی می کنیم.

با استفاده از مسيردهی در یک برنامه تک صفحه ای می توانیم بخشی از یک آدرس را گرفته و آن را با آدرس دیگری جایگزین کنیم. با این روش می توانیم به طور دینامیک در مرورگر بین آدرس های مختلف جابجا شویم.

به بیان دیگر، یک کامپوننت ری اکت را به عنوان کامپوننت ریشه (root) برای عناصر داخل برنامه تعریف می کنیم. سپس می توانیم به ری اکت بگوییم که با جایگزین کردن یک کامپوننت به جای دیگری، نمای ظاهری یک ویو را به طور کلی تغییر دهد.

کامپوننت App را باز کرده و تمام روت های مورد نیاز را در این کامپوننت تعریف می کنیم. همچنین باید تعدادی کامپوننت را از پکیج react-router گرفته و به برنامه وارد کنیم. شکل (2-8) نمونه ای از استفاده از این کامپوننت ها را نشان داده است.

10-7-2- تست نرم افزار

هنگامی که از تست صحبت می کنیم، منظور ما پیاده سازی یک سرویس منظم برای جلوگیری از بروز خطا در نرم افزارها است. یعنی به جای اینکه نرم افزار را در محیط هاست و سرور بررسی کنیم، در حالت لوکال (داخلی) می توان تمام ایرادهای آن را برطرف کرد.

هنگامی که درباره تست front-end صحبت می کنیم، در حقیقت منظور ما، تست اینکه برنامه ری اکت ما چه چیزی را رندر می کند و همچنین نحوه پاسخ آن به تعامل کاربر می باشد.

در ادامه درباره سه الگوی مختلف تست نرم افزار صحبت می کنیم: تست واحد، تست عملکرد و تست یکپارچگی.

- تست واحد

در تست واحد (unit testing)، هر بخش اختصاصی از کدهای برنامه را به صورت مجزا تست می کنیم تا مطمئن شویم آن قسمت از کد مطابق انتظار ما کار می کند.

برای مثال ما قبلاً چند reducer در برنامه مان ایجاد کرده بودیم. هر کدام از این Reducer ها یک کار واحد را انجام می دهند، حال با کمک تست واحد می خواهیم ببینیم که آیا این reducer ها تحت سناریو های مختلف به خوبی کار می کنند یا خیر. در ری اکت، تست واحد به مرورگر نیاز ندارد و می توان با سرعت بسیار بالاتری آنرا اجرا کرد و فرآیند تست هم بسیار ساده انجام می شود.

در اکثر مواقع باید به جواب این سوال تمرکز کنیم: با گرفتن یک سری ورودی (state) و prop ها، آیا خروجی با انتظارات ما از چیزی که باید در dom مجازی نمایش داده شود، مطابقت می کند؟ در این مورد، ما نمایش خروجی را تست می کنیم.

- تست عملکرد

در تست عملکرد ما روی تست رفتار کامپوننت مان تمرکز می کنیم. برای مثال، اگر یک نوار پیمایش (navigation bar) با یک دکمه login / logout داشته باشیم، توقع داریم که: اگر کاربری به برنامه لاگین کرد، نوار پیمایش دکمه logout را نمایش دهد. اگر کاربر لاگین نکرده بود، دکمه login به او نمایش داده شود.

تست عملکرد معمولاً به صورت مجزا اجرا می شود (برای مثال تست عملکرد کامپوننت، بدون تست بقیه برنامه)

- تست یکپارچگی

در انتها آخرین تستی که بررسی می کنیم، تست یکپارچگی است. این نوع تست، کل سرویس های برنامه را تست کرده و سعی دارد همان تجربه ای که کاربر نهایی قرار است داشته باشد را به ما در هنگام توسعه برنامه بدهد. در مورد سرعت و کارایی، تست یکپارچگی سرعت بسیار پایینی دارد، چرا که باید درخواست های ما را در یک مرورگر اجرا کند، به همین دلیل تست های واحد و عملکردی با سرعت بالاتری از تست های یکپارچگی اجرا می شوند (در ری اکت تست عملکردی در یک dom مجازی به جای مرورگر واقعی اجرا می شود، در نتیجه سرعت بالاتری دارد)

هنگام تست کامپوننت های ری اکت هم باید ببینیم که در dom مجازی چه چیزی قرار گرفته و همچنین چه چیزی در dom واقعی نمایش داده خواهد شد.

```
import React from 'react';
import { MemoryRouter, Route } from 'react-router-dom';
import mockedAxios from 'axios';
import { cleanup, fireEvent, render, screen, waitFor } from '@testing-library/react';
import Login from '../Login';
import { API_LOGIN } from '../apiConstants';

afterEach(cleanup);
jest.mock('axios');

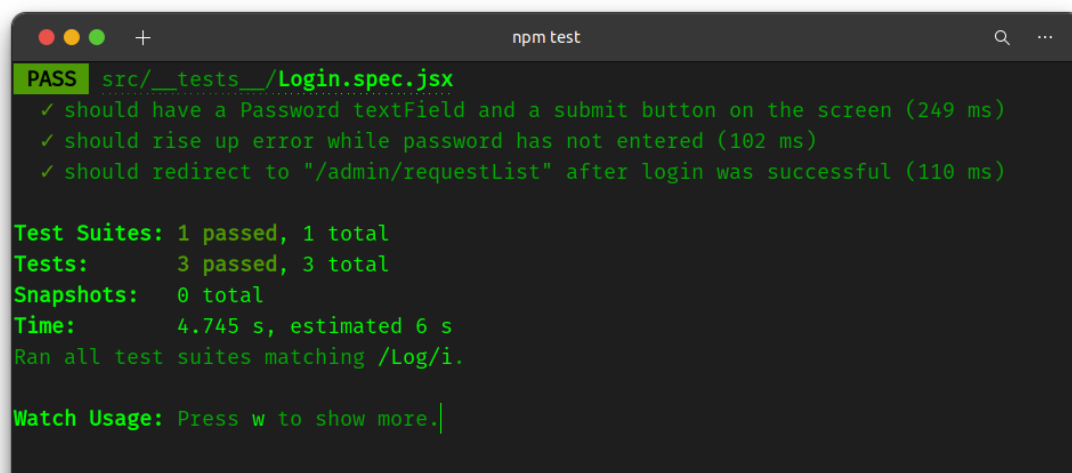
it('should redirect to "/admin/requestList" after login was successful', async () => {
  mockedAxios.post.mockResolvedValueOnce({
    status: 200,
    data: [{
      token: 'Test Token',
    }],
  });
  render(
    <MemoryRouter>
      <Login />
      <Route path="/admin/requestList">request list page</Route>
    </MemoryRouter>,
  );
  const password = screen.getByLabelText('رمز عبور');
  fireEvent.change(password, { target: { value: 'Test Password' } });
  const submit = screen.queryByText('ارسال');
  await waitFor(() => expect(submit).toBeInTheDocument());
  fireEvent.click(submit);
  await waitFor(() => {
    expect(mockedAxios.post)
      .toHaveBeenCalledWith(API_LOGIN, { password: 'Test Password' });
    expect(screen.queryByText('request list page')).toBeInTheDocument();
  });
});
```

شکل (2-16) نمونه تستی که برای کامپوننت Login نوشته شده است.

برای اجرای تست ها کافیسست دستور زیر را در ترمینال وارد کنیم :

```
npm test
```

که به طور مثال خروجی اجرای تست کامپوننت Login (شکل (2-16)) به صورت زیر است :



```
npm test
PASS src/__tests__/Login.spec.jsx
  ✓ should have a Password textField and a submit button on the screen (249 ms)
  ✓ should rise up error while password has not entered (102 ms)
  ✓ should redirect to "/admin/requestList" after login was successful (110 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        4.745 s, estimated 6 s
Ran all test suites matching /Log/i.

Watch Usage: Press w to show more.
```

شکل (2-17) نمونه خروجی اجرای تستی که برای کامپوننت Login نوشته شده است

8-2- آشنایی با کتابخانه Material UI [10]

Material ui هم یکی از معروف ترین کتابخانه های material design هست که برنامه نویسی ریکت رو برای توسعه دهنده ها خیلی راحت تر کرده . شما با استفاده از Material-UI ، میتونین خیلی راحت از عناصر material design در برنامه ای که با ریکت مینویسین استفاده کنین Material ui هم مثل ریکت، کامپوننت بیس هست و مجموعه ای از کامپوننت های React است که برای سهولت طراحی رابط کاربری استفاده میشه. در ضمن استفاده از material ui خیلی راحت و با پکیج های npm قابل نصب و استفاده ست.

یکی از مزایایی که داره اینه که برای استفاده لازم نیست کل کتابخونه رو دانلود کنید و فقط هر بخشی که دوست دارین رو دانلود و استفاده میکنید.

به عنوان مثال برای اینکه شما بتونین به آیکون رو توی برنامه تون بیارین میتونین از نمونه هایی که وجود داره یکی رو انتخاب کنین و توی برنامه تون import کنین.

نسخه نمایشی (Demo) تمام کامپوننت ها رو میتونین قبل از اضافه کردن به برنامه تون ببینید.

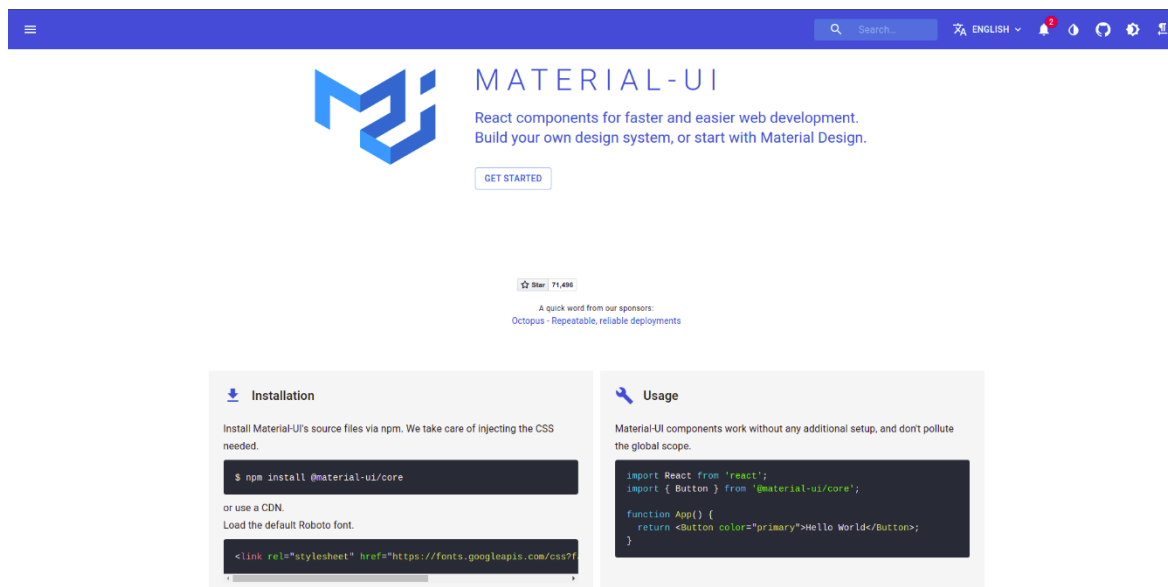
جدول، فرم ، header ، bottom ، footer و هر چیزی که شما توی ظاهر یک سایت یا اپلیکیشن میبینین رو میتونین با استفاده از کامپوننت های متریال یو آی به برنامه تون اضافه کنین .

با توجه به اینکه در حال حاضر رسپانسیو بودن یک برنامه یکی از مهمترین ویژگی هاییه که به برنامه باید داشته باشه با استفاده از متریال یو آی میشه به برنامه رسپانسیو رو ایجاد کرد.

یکی از مسائلی که توی برنامه نویسی برای برنامه هایی با زبان فارسی است اینه که لازمه RTL باشه و متریال یو آی همچنین امکانی رو خیلی راحت برای توسعه دهنده ریکت فراهم میکنه .

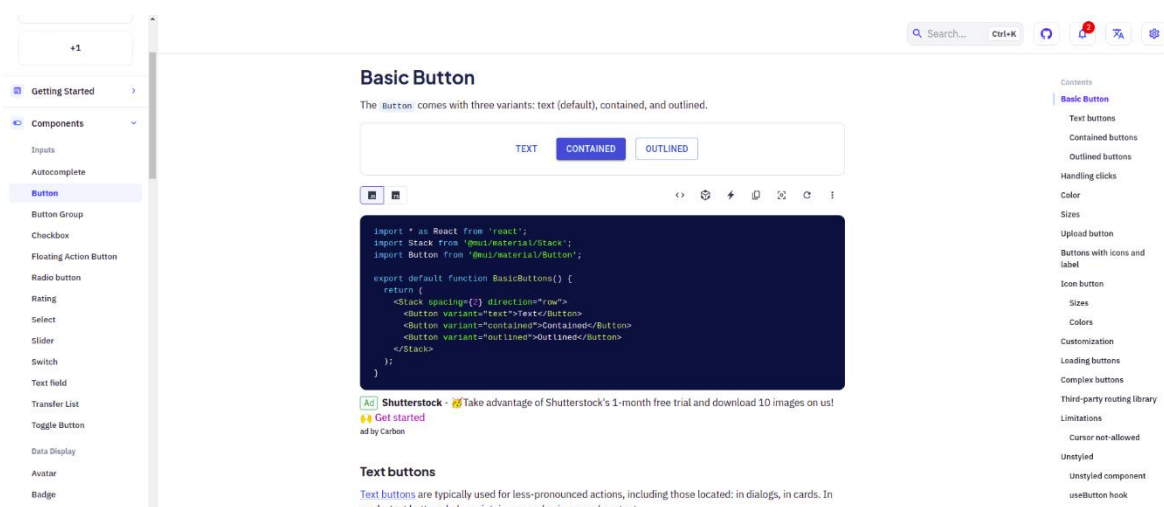
اگر یکی از کامپوننت هایی که لازم داشتن ولی لازم بود تغییراتی در اون ایجاد بشه، متریال یو آی امکان سفارشی سازی کامپوننت ها رو هم براتون فراهم میکنه .

برای نصب این کتابخانه کافیه به سایت آن سرزنید و طبق داکيومنت های آن عمل کنید.



شکل (2-18) صفحه دانلود کتابخانه mui از سایت اصلی آن

به عنوان مثال ، من نمونه از داکيومنت این کتابخانه برای استفاده از Button آن را به صورت عکس آورده ام.



شکل (2-19) بخشی از داکيومنت کتابخانه mui مربوط به استفاده از Button

9-2- معرفی کلیتی از پروژه Premium TradingView

نیاز اصلی که منجر به ایجاد این پروژه شده است ، بحث تحریم های بانکی و مالی علیه ایران است که موجب شده تا ایرانیان برای خریدهای بین المللی و انتقال پول دچار مشکل باشند.

TradingView از مرورگرهای بر پایه پلتفرم های نموداری است. سایت تریدینگ ویو به عنوان دستیار تریدر در تحلیل و بررسی راحت تر چارت به او کمک میکند. این مرورگر می تواند با در اختیار قرار دادن طیف وسیعی از

ابزارها برای معامله گران فارکس و یکپارچه سازی نمودارها، استاندارد صنعت استفاده از نمودارها را ارتقا بخشد. برخی از کارگزاران فارکس، نرم افزارهای نموداری را ارائه می دهند که معمولاً همراه با گزینه های سفارشی محدود هستند. لذا سایت تریدینگ ویو میتواند پلتفرم کامل و جامع تری نسبت به متاتریدر بروکرها برای تحلیلگران تکنیکال است بازار فارکس باشد.

TradingView پلتفرمی است که در اختیار تمام معامله گران فارکس قرار دارد. می توانید در این پلتفرم با سایر معامله گران درباره موضوعات مختلف، بحث و تبادل نظر داشته باشید، از نظرات آنها و اجتماعات مختلف افرادی که در این پلتفرم عضویت دارند، استفاده کنید. البته که بکارگیری این پلتفرم برای معامله گران هزینه بر است اما این امر برای معامله گرانی که به صورت جدی وارد عرصه تجارت شده اند، نوعی سرمایه گذاری به حساب می آید.

تنها در مواردی مانند مشاهده ی تایم فریم هایی غیر از تایم فریم های پیش فرض و استفاده ی همزمان بیشتر از سه اندیکاتور/اسیلاتور یا ست کردن چندین هشدار به طور همزمان و برخی امکانات دیگر نیاز به استفاده از نسخه های Pro و Premium است که مبلغی به صورت ماهیانه دریافت می کند.

به دلیل تحریم های بانکی و مالی، خرید اکانت های pro و premium برای کاربران ایرانی، تقریباً غیرممکن شده است لذا همین نیاز منجر به پیدایش این سایت PremiumTradingView شده است.

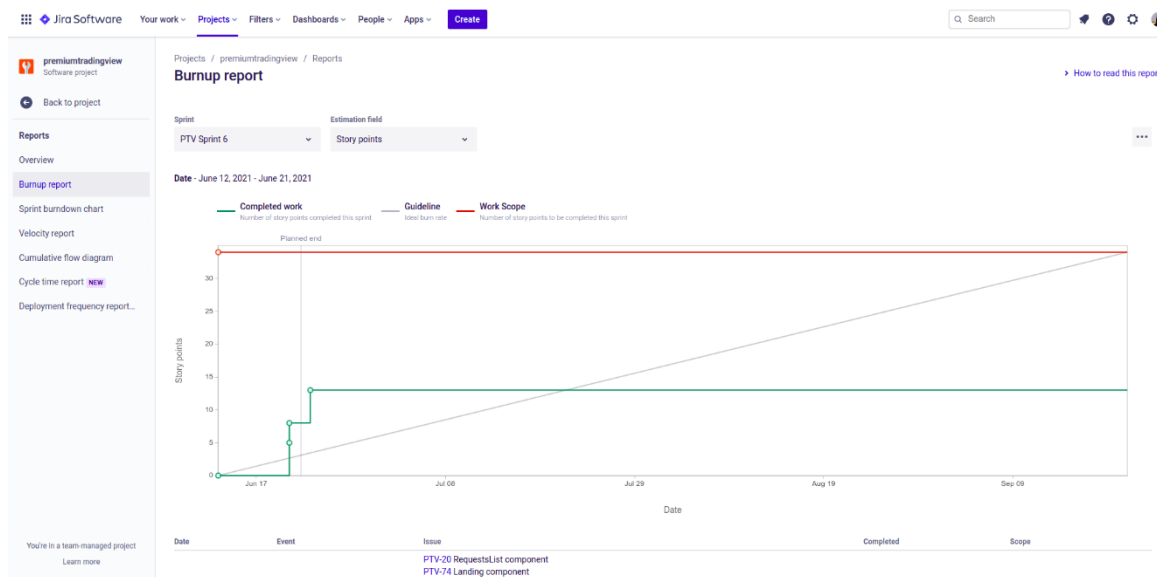
این پروژه در یک نسخه mvp آماده و قابل عرضه به مشتریان شده است (هرچند که مراحل مالیاتی و کسب نشان اعتماد آن باقیمانده است).

در نسخه mvp این پروژه، یک داشبورد مدیریت داریم که به درخواست های کاربران (مبنی بر اکانت های pro, pro+ و premium) را مشاهده و به آنها رسیدگی می کند (مبلغ پرداخت شده توسط کاربران را تایید و معادل دلاری آن را به سایت اصلی tradingview.com پرداخت می کند و اکانت مورد نظر کاربر را برایش فعال می کند) و یک صفحه Login و یک صفحه خرید اکانت می باشد.

در پیاده سازی نسخه mvp این پروژه، یک تیم دو نفره متشکل از بنده به عنوان فرانت اند کار و یک کارآموز دیگر به عنوان بک اند کار مشارکت داشته اند.

پیاده سازی نسخه mvp این پروژه ۶ اسپرینت زمان برده است که در ادامه شکل هایی از صفحه مدیریت پروژه جیرا به همراه یک نمونه نمودار برن آپ آورده ام.

شکل (2-20) بورد جیرا مربوط به اسپرنت ششم پروژه premiumTradingView



شکل (2-21) Burnup chart مربوط به اسپرنت ششم پروژه premiumTradingView

مراجع

- <https://zamin.link> [1]
- <https://balonet.net> [2]
- <https://www.w3schools.com/html> [3]
- <https://www.w3schools.com/css/default.asp> [4]
- <https://www.atlassian.com/agile/tutorials> [5]
- <https://git-scm.com/docs/gittutorial> [6]
- <https://www.w3schools.com/js/default.asp> [7]
- <https://www.w3schools.com/react/default.asp> [8]
- <https://www.roxo.ir/series/react-tutorials> [9]
- <https://mui.com/getting-started/installation> [10]