

دانشگاه فنی و حرفه ای

دانشکده فنی و حرفه ای دختران دکتر شریعتی

پایان­نامه کارشناسی ناپیوسته

رشته کامپیوتر گرایش نرم افزار

برنامه ریز(پلنر)

نگارش:

زهرا عربیان حسین آبادی

استاد راهنما:

سمانه جان آقایی

مرداد1401



چکیده

آیا تا به حال به این فکر کرده اید برای پس‌انداز خود چه برنامه ای داشته باشید؟یا برای اهدافی که در نظر دارید چه برنامه ای نیاز دارید تا به صد‌در‌صد موفقیت برسید؟

با برنامه ریزی شما خواهید دانست که وقت خود را بیهوده صرف نکنید و کنار تفریح خود به انگیزه‌ای که دارید برسید.این انگیزه میتواند کم کردن وزن یا پس‌انداز و حتی تحصیل باشد.

با برنامه ریزی، به مسیر درست دست خواهید یافت. به این شکل، شما حتی اگر مدتی از اهداف و آرزو های خود دور باشید و آنها را به فراموشی بسپارید، با مرور برنامه خود دوباره به مسیر هدف‌تون باز خواهید گشت.

بااین وجود، در این پژوهش سعی کردیم روشی را برگزینیم تا با آن افراد بتوانند برنامه روزانه ای برای خود تنظیم کنند، آنهارا ثبت کنند و به اهداف خود برسند.

همچنین، برای آن نمایی با مفاهیم ui/ux طراحی کرده ایم تا انگیزه بیشتری برای افراد ایجاد کنیم.

در ادامه آنها را در قالب تحت وب ارائه کردیم تا در هر زمان برای هر فرد قابل دسترس باشد.

**کلیدواژه:** پلنر،برنامه ریز، برنامه ریزی، هدف، وب اپلیکیشن، فرونت-اند

فهرست نوشتار

[فصل 1: مقدمه 1](#_Toc57377650)

[1-1- انگیزه 2](#_Toc57377651)

[1-2- مروری بر پیشینه و کارهای مشابه 2](#_Toc57377652)

[1-3- هدف 2](#_Toc57377653)

[1-4- رئوس مطالب سایر فصل‌ها 2](#_Toc57377654)

[فصل 2: تجزیه و تحلیل نیازمندی‌ها 4](#_Toc57377655)

[2-1- مقدمه 5](#_Toc57377656)

[2-2- نتیجه‌گیری 6](#_Toc57377657)

[فصل 3: ساختار داده‌ها و بانک اطلاعات 7](#_Toc57377658)

[3-1- مقدمه 8](#_Toc57377659)

[3-2- طراحی 8](#_Toc57377660)

[3-3- نرم‌افزار مدیریت پایگاه داده‌ها 8](#_Toc57377661)

[3-4- نتیجه‌گیری 8](#_Toc57377662)

[فصل 4: پیاده‌سازی 9](#_Toc57377663)

[4-1- مقدمه 10](#_Toc57377664)

[4-2- کدنویسی 10](#_Toc57377665)

[4-3- اجرای نرم‌افزار 10](#_Toc57377666)

[4-4- نتیجه‌گیری 11](#_Toc57377667)

[فصل 5: جمع‌بندی و پیشنهادها 12](#_Toc57377668)

[5-1- نتیجه‌گیری 13](#_Toc57377669)

[5-2- پیشنهادهایی برای کارهای آتی 13](#_Toc57377670)

[منابع 15](#_Toc57377671)

[پیوست‌ها 17](#_Toc57377672)

فهرست شکل­ها(در صورت وجود)

[شکل 2-1. نمونه شکل 5](#_Toc57227971)

[شکل 4-1. زیرنویس شکل 11](#_Toc57227972)

[شکل 5-1. زیرنویس شکل 14](#_Toc57227973)

فهرست جدول­ها (در صورت وجود)

[جدول 2-1. نتیجه بررسی پرسش نامه ها در ارتباط با عوامل موثر 5](#_Toc57227919)

[جدول 4-1. بالانویس جدول 11](#_Toc57227920)

[جدول 5-1. بالانویس جدول 13](#_Toc57227921)

1. مقدمه
   1. انگیزه

از دیر‌با‌ز اکثر افراد موفق، برنای رسیدن به اهداف خود برنامه ریزی می‌کردند و تا دستیابی به آنها مسیر را بررسی می‌کردند تا نقص ها و موانع را برطرف کرده و موارد پیشرفت را در آن بیفزایند.

امروزه،هر شخصی در هر جایگاهی، بخصوص در جوان ها و نوجوان ها که در ابتدای آینده خود هستند،از برنامه ریزی استفاده می‌کنند تا از فرصت های خود به خوبی بهره‌مند شوند.

* 1. مروری بر پیشینه و کارهای مشابه

ساده ترین روش برنامه ریزی، استفاده از قلم و کاغذ است.

با گذشت زمان و فراگیر شدن ایده برنامه ریزی، صنعت چاپ کاغذ ها و دفاتری را ارائه نمود تا انگیزه بیشتری برای افراد ایجاد کند و هم زمان کمتری صرف طراحی و هدر رفت کاغذ شود.

به این صورت موضوعاتی ارائه شد تا همه افراد امکان استفاده را داشته باشند.

این برنامه ریز های دفتری، به صورت برنامه درسی، کارهای روزانه، اهداف، و کارهای شخصی ایجاد شد تا هر شخص متناسب با هدف برنامه ریزی خود را انجام دهد.

پیرو ایده برنامه ریزی، برنامه نویسان راهکاری را پیشنهاد دادند تا هم در مصرفه کاغذ صرفه جویی شود و هم در هر زمان و مکانی قابل استفاده باشد.

به این صورت اپلیکیشن های موبایلی ارائه شد تا افراد از طریق اپلیکیشن ها به مسیر پیشرفت خود را مرور کنند.

* 1. هدف

بسیاری از افراد هستند از موبایل ها استفاده کمتری می‌کنند و یا شاید برای افرادی پیش بیایید که موبایل آنها به هر دلیلی از دسترس خارج شود و یا حتی حافظه کافی برای اپلیکیشن های موبایلی نداشته باشند.

بنابراین، در این تحقیق سعی کردیم روشی را ارائه دهیم تا هر شخص بتواند در هر زمان برای آینده خود برنامه ریزی داشته باشد.

به این صورت، برنامه تحت وب ارائه داده ایم که از طریق سایت های اینترنتی، هم از طریق تلفن همراه و هم از طریق سیستم های کامپیوتری قابل استفاده باشد.

به این صورت افراد از طریق برنامه های تحت وب[[1]](#footnote-1) بدون نیاز به نصب هیچ برنامه‌ای، می‌توانند برای خود برنامه شخصی ایجاد کنند.

* 1. رئوس مطالب سایر فصل‌ها

معرفی محتوای سایر فصل­ها و ساختار پایان­نامه

1. تجزیه و تحلیل نیازمندی‌ها
   1. مقدمه

همانطور که در فصل اول بیان کردیم برای شروع برنامه ریزی تنها به یک قلم و کاغذ احتیاج است؛ اما اگر زمان کافی برای بیان نیاز های خود ندارید یا نمی‌دانید از کجا شروع کنید و چطور آنها را بیان کنید به شما راهکار استفاده از روش های الکترونیکی را پیشنهاد می‌دهیم. به این صورت شما از قالب های آماده ای که بر اساس نیاز شما طراحی شده است استفاده می‌کنید و برنامه خود را بر روی آنها ثبت می‌کنید تا هر زمانی که نیاز به مرور داشتید به آنها رجوع کنید.به این صورت، دیگر نگران فراموشی آنها نیستید و باخیال راحت روی موارد بیشتر تمرکز می‌کنید.

بعلاوه راهکاری ارائه کردیم تا با آن بتوان در هر شرایطی برنامه ریزی کرد.

به عنوان مثال تلفن های همراه امروزه، پس از یک بازه زمانی نیازمند به روز رسانی هستند؛ به این ترتیب باید اپلیکیشن [[2]](#footnote-2)هایی نوشت تا با آنها مطابقت داشته باشند. بعلاوه حجمی از حافظه دستگاه را اشغال می‌کنند.از موارد دیگری که می‌توان اشاره کرد، پرداخت درون برنامه‌ای و تبلیغات هستند که در این برنامه ها قرار دارد.

از این‌رو، نرم افزار‌های تحت وب را پیشنهاد می‌دهیم که تا افراد از طریق مرورگر‌ها و بدون نیاز به نصب، این نرم افزار‌ها را در اختیار داشته باشند.بعلاوه، با استفاده از نسخه تحت وب شما قادر خواهید بود با سایر دستگاه‌ها که بارز‌ترین آنها کامپیوتر‌ها و لب‌تاب‌ها هستند استفاده کنید.

در ادامه این فصل، قصد داریم درباره خصوصیات یک وب اپلیکیشن صحبت کنیم.

وب اپلیکیشن چیست و چگونه می‌توان با آن برنامه ریزی کرد، این برنامه‌ریزی چگونه خواهد بود وچطور طراحی شده است.

همانطور که می‌دانید حوزه برنامه نویسی به چند دسته تقسیم می‌شود و یکی از حوزه‌های آن برنامه نویسی تحت وب است.زیر‌شاخه‌های برنامه نویسی تحت وب به دو دسته فرونت-اند [[3]](#footnote-3)و بک-اند[[4]](#footnote-4) تقسیم می‌شود که در بخش در بخش بک-اند کدنویسی هایی صورت می‌گیرد که به راحتی برای کاربر قابل دسترس نیست و روی سرور پردازش می‌شوند.اما برنامه نویس در سمت فرونت-اند بر روی ظاهر یک سایت تسلط دارد و بخشی را آماده مي‌کند که مستقیما در دید کاربر است و باآن ارتباط سمعی و بصری دارد.

هر کدام از این شاخه‌ها، از زبان‌های برنامه نویسی استفاده می‌کنند؛به عنوان مثال در بخش بک-اند از زبان برنامه نویسی سی شارپ[[5]](#footnote-5)، پی اچ پی[[6]](#footnote-6)، پایتون [[7]](#footnote-7)و... استفاده می‌شود و در بخش فرونت-اند از جاوا اسکریپت[[8]](#footnote-8) استفاده می‌شود که لازم به ذکر است در این پروژه از این زبان و کتابخانه‌های [[9]](#footnote-9)آن استفاده شده است.بعلاوه، لازم است بیان دارم اگر تاکنون html [[10]](#footnote-10)را زبان برنامه نویسی تصور می‌کردید در اشتباه بوده‌اید.

HTML تنها زبان نشانه گذاری برای مفاهیم اینترنت است و اصولا اسکلت یک سایت را تشکیل می‌دهد و با کمک CSS [[11]](#footnote-11)به آن ظاهر قابل درک ارائه می‌دهیم.بنابراین زبان برنامه نویسی که به آن احتیاج است تا سایت به خواسته ما نزدیک شود،زبان جاوا اسکریپت است.

با کمک جاوا اسکریپت، ظاهر سایت را می‌توان به گونه ای برنامه ریزی کرد که از حالت ایستا به پویا تبدیل شود.بدین ترتیب، کاربران می‌توانند با سایت تعامل داشته باشند و فقط خواننده یک مطلب از وبلاگ ها نباشند.

بدین شکل، وب اپلیکیشن ها به بازار ارائه می‌شوند تا نه تنها تعامل پویا ایجاد کنند، بلکه قابلیت های جدیدی به خدمات سایت اضافه کنند.

به عنوان مثال، سایت گوگل را در نظر بگیرید؛علاوه بر خدمات جستجو چه قابلیت دیگری برای خدمت رسانی ارائه کرده است؟

یکی از پرمخاطب ترین این خدمات،پیام رسان جی میل[[12]](#footnote-12) است؛که به کاربران خود امکان پیام رسانی را هم از طریق سایت و هم از طریق نرم افزار‌های موبایلی ارائه کرده است.

به این ترتیب، این خدمت از طریق سایت وب اپلیکیشن نام دارد.بسیاری از افراد از طریق وب اپلیکیشن جی-میل، نامه های الکترونیکی خود را چک می‌کنند.همچنین این نسخه های تحت وب با نسخه های موبایلی خود همگام هستند که این بخش نیازمند برنامه نویسان بک-اند می‌باشد.

در این پروژه، ما هم سعی کرده‌ایم با استفاده از روش های برنامه نویسی فرونت-اند، وب اپلیکیشنی را ارائه کنیم تا کاربران بتوانند به آن دسترسی داشته باشند و رخدادهای روزمره خود را با ایده های نو در آن ثبت کنند.

از این رو، در این پروژه مواردی را که مورد توجه بیشتر کاربران است، با استفاده از قواعد طراحی ui[[13]](#footnote-13)/ux[[14]](#footnote-14)

در قالب SPA [[15]](#footnote-15) پیاده سازی کرده‌ایم.به این صورت کاربر پس از وارد شدن به سایت اصلی و وارد کردن اطلاعات شخصی در فرم، وارد صفحه برنامه ریزی شده و با انتخاب بخش مورد نظر وارد جزئیات آن می شود.

این موارد عبارتند از :

1. مقدار آب مصرفی که هرشخص در طول روز میل می‌کند.

دلیل انتخاب این مورد به این خاطر است که، افراد برای حفظ سلامت و تندرستی خود نیاز دارند آب مورد نیاز بدن را تامین کنند.طبق تحقیق دانشمندان هر شخص تا 7 لیوان در روز می‌تواند آب میل کند.که این رقم وابسته به سوخت و ساز بدن است.

بعلاوه، افرادی که یک روتین ورزشی رو دنبال می‌کنند مقدار آب بیشتری مصرف می‌کنند و به این شکل سوخت و ساز بدن بالا رفته و به کاهش وزن کمک می‌کند.

1. داروهایی که هر فرد به آنها نیاز دارد.

مصرف دارو لزوما به این خاطر نیست که شخصی بیمار است؛ بلکه برخی افراد متناسب با شرایط خود برخی داروهای مکمل یا تقویتی مثل ویتامین ها را مصرف کند یا ممکن است فردی از قرص‌های خواب آور استفاده کند.از این رو این بخش را برای این افراد در نظر گرفته ایم تا مصرف خود را در نظر داشته باشند.

1. هر شخصی در طول روز یک احساس درونی در خود حس می‌کند که می‌تواند روی بدن و کارهایی که انجام بدهد تاثیرگذار باشد.به عنوان مثال فرض کنید شخصی که از ابتدای شروع روز خود عصبانی یا نگران باشد چه تاثیری روی سایر دارد؟

بنابراین، با این ایده هر فرد در پایان روز یا هفته تاثیرات احساسات خود را بر روی کارهایی که انجام می‌دهد مرور می‌کند و بر احساسات خود کنترل بیشتری پیدا خواهد کرد.

1. زمانی که فرد برای خود هدفی را تعیین می‌کند یعنی بیهوده زندگی نمی‌کند و این تعیین هدف به او انگیزه می‌دهد تا از تنبلی دور باشد.بنابراین خوب است فرد اهدافی را در مراحل مختلف زندگی داشته باشد و پس از رسیدن به آن خود را ارزیابی کند.
2. هدف بزرگ انسان سعادت است و به پاس این سعادت و برای هر امتحان زندگی باید خدارا شکر گفت.در این پروژه چالشی با عنوان تشکر از خدا ایجاد کردیم تا یاد خدای بزرگ فراموش نشود.
3. هر شخصی در روز یک‌سری کار شخصی برای انجام دارد، همانند مطالعه، تفریح، نکات بهداشتی و... که می‌توان آنها را یادداشت کرد تا به ترتیب به آنها رسیدگی شود.
4. حتما هر فردی یک لیست از نیاز‌های خود تهیه می‌کند و برای آن الویت بندی می‌کند که در چه بازه زمانی خریداری شود.
5. به عنوان یک ایده جدید، برای کاربران بخش فیلم و سریال قرار داده ایم تا با جستجوی فیلم مورد نظر اطلاعات آن را در اختیار داشته باشد و در اوقات فراغت خود از دیدن فیلم یا سریال لذت ببرد.
6. بعضی از موارد لازم است ثبت شود تا و نیاز به گذر ندارد.پس در بخش رخداد ها شرح روز را نگه دارید.که این امکان را باثبت موارد گفته شده قرار داده ایم.

که در هرکدام از این موارد جزئیاتی قرار داده‌ایم تا بتوان این موارد را به کار برد.

این جزئیات شامل اینپوت‌ها[[16]](#footnote-16)، دکمه ها، نمادهاست تا کاربران را در جریان کار قرار دهد.

البته فقط قرار دادن موضوع به تنهایی کافی نیست بلکه همانند یک ساختمان که اسکلت آن آماده شده است نیازمند ظاهری جذاب برای کاربر است.بنابراین، در این پروژه از نمادهای متحرک، پس زمینه جذاب، فرم هایی که به صورت مُدال[[17]](#footnote-17) قرارا گرفته اند استفاده نموده ایم.

با استفاده از جمله های انگیزشی کوتاه سعی در مغلوب کردن کاربر داریم.در اینجا قواعد ui/ux به کمک ما آمدند تا ظاهر را اصولی طراحی کنیم.

­­

* 1. نتیجه‌گیری

در این فصل سعی کردیم موارد لازم در مورد طراحی یک بستری که بتوان برنامه ریزی روزانه را ساده‌تر کند صحبت کنیم. این بستر یک وب سایت با محتوای ساده برای ارائه یک خدمت اینترنتی به صورت وب اپلیکیشن پیشنهاد شد؛ که در این وب اپلیکیشن، مواردی قرار گرفت تا یک کاربر بتوانند احساسات خود را بیان کند، کارهای شخصی خود را به موقع انجام دهد و آنرا ثبت کند.همچنین برای رفاه حال کاربر ایده‌ای برای فیلم وسریال و تقویم برای او قرار داده شده است.

همچنین بیان کردیم که یک وب اپلیکیشن همانند یک ساختمان نیازمند ظاهری جذاب است که با استفاده از قواعد ui/ux آنرا طراحی کردیم.

این طراحی شامل موارد بیان احساسات، مقدار آب مصرفی، داروهای مصرفی، هدف های یک فرد، لیست خرید و لیست کارهای شخصی، فیلم و سریال و تقویم، و چالش شکر گذاری خدا که هر کدام جزئیاتی از جمله اینپوت‌ها، دکمه ها، نمادهاست تا کاربران را در جریان کار قرار دهد.

در فصل های آینده چگونگی طراحی و ساختار آن بیان خواهیم کرد که چطور این وب اپلیکیشن شکل می‌گیرد.

­­­­

1. ساختار داده‌ها و بانک اطلاعات
   1. مقدمه

در فصل قبل، درباره وب اپلیکیشن صحبت کردیم و گفتیم برای طراحی یک وب اپلیکیشن به چه مواردی نیاز داریم.

در این فصل بیان کردیم از وب اپلیکیشن برای ارائه یک برنامه پلنر به صورت آنلاین بر روی سایت استفاده می‌شود و هر المان در برنامه را شرح دادیم.

در فصل سوم، قصد داریم نحوه شکل گیری یک وب اپلیکیشن و منابعی که برای ذخیره اطلاعات استفاده می‌شود صحبت کنیم.همچنین درباره جزئیات یک برنامه پلنر بحث خواهیم کرد.چطور یک سایت به وب اپلیکیشن تبدیل می‌شود و چگونه می‌توان وارد برنامه پلنر شد؟

در ادامه خواهیم دید یک وب اپلیکیشن چگونه طراحی می‌شود و از چه منابعی استفاده می‌کند.

* 1. طراحی

در ابتدا برای داشتن یک وب سایت نیاز است از کدهای html استفاده نماییم و اسکلت یک سایت را بنا کنیم.سپس با استفاده از کدهای css ، برای سایت نمای قابل درک ایجاد می‌کنیم.

از کدهای جاوااسکریپت برای پویا کردن سایت کمک می‌کنیم که کتابخانهReact آن در این پروژه مورد استفاده قرار گرفته است.

پس از طراحی سایت و بیان منظور، نیاز به اعتبار سنجی افراد داریم. بنابراین،فرم هایی طراحی کرده تا اطلاعاتی از کاربر گرفته و در صورت صحت اطلاعات وارد حساب شود.

پس از ورود، به بخش اصلی پروژه، یعنی وب اپلیکیشن خواهیم رسید.

در این وب اپلیکیشن با استفاده از کتابخانه جاوااسکریپت،وب اپلیکیشین را طراحی می‌کنیم.

در ری اکت،با استفاده از کدهای html و دستورات جاوااسکریپت، طراحی وب را در داخل کامپوننت‌ها(در فصل آینده توضیح داده می‌شود) قرار می دهیم.

از این کامپوننت‌ها، در تمامی مراحل پروژه استفاده شده است.سپس با استفاده از روترها [[18]](#footnote-18)به یکدیگر لینک [[19]](#footnote-19)داده شده اند.

در برخی از کامپوننت‌ها نیاز به استفاده از کتابخانه های جاوااسکریپتی داشته‌ایم؛پس آنها را به پروژه افزوده و ذخیره کرده‌ایم.

در برخی دیگر از کامپوننت‌ها، نیاز به منبعی داریم تا داده ها به طور دائمی در آن قرار گیرند؛

بنابراین از localstorage و axios استفاده کردیم که در ادامه توضیح خواهیم داد.

* 1. نرم­افزار مدیریت پایگاه­ داده­ها

با استفاده از کدهای جاوااسکریپتی داده های مورد نیاز از جمله اطلاعات کاربر را در مرورگر ذخیره کرده‌ایم تا با هربار ورود، اطلاعات از بین نرود.

از localstorage در موارد دیگر از جمله موارد shoplist استفاده نموده ایم.

برا یاستفاده از localstorage نیاز به نصب هیچ کتابخانه و یا برنامه اضافی نیست؛ بلکه کافی است از JSON [[20]](#footnote-20)برای خواندن و قرار دادن داده ها استفاده می‌شود.بنابراین، از دستور زیر برای قرار دادن اطلاعات در localstorage مرورگر استفاده می‌کنیم:

localStorage.setItem('Do',JSON.stringify(this.state.tempinput));

در کد بالاsetitem مقدار مورد نظر را در localstorage قرار می دهد.

‘Do’ یک کلید برای داده در نظر می‌گیرد و داده tempinput را درون آن قرار می‌دهد.

دستورstringify داده را از نوع object به string تبدیل می‌کند.

بدین صورت محتوای مورد نظر در حافظه مرورگر ذخیره می‌شود.

سپس با دستور زیر آنرا از حافظه فراخوانی می‌کنیم:

const data = localStorage.getItem('Do');

برای نگه داری داده و نمایش آن نیاز به تعریف متغیر داریم که در اینجا از نوع ثابت آن را تعریف کرده‌ایم.

متغیر ما حاوی اطلاعاتی با کلید’Do’ می باشد که با دستور getitem از حافظه مرورگر فراخوانی می‌شود.در این متغیر مقدارtempinput که در حافظه مرورگر ذخیره کردیم قرار گرفته است و حالا در متغیر data قرار می‌گیرد.

برای نمایش مقدار این متغییر از دستوراتجاوااسکریچتی برای نمایش object ها به صورت زیر استفاده می‌کنیم:

{data}

سپس کد بالا را در میان کدهای html درست جایی که قرار است داده ها نمایش یابند قرار می‌دهیم.

Axios یک api[[21]](#footnote-21) برای ذخیره و نگه‌داری اطلاعات است که باید به صورت دستوری در command line یا ترمینال برنامه به صورت زیر نصب شود:

npm install axios

و با –save آن را در پروژه خود ذخیره می‌کنیم.این api در package.json پروژه ذخیره می‌شود و در پروژه اعمال می‌شود.اما برای استفاده از آن لازم است آن را در جای پروژه که قرار است استفاده شود import کرد.

بنابراین با دستور زیر آنرا به محل استفاده در پروژه اضافه خواهیم کرد:

Import axios from ‘axios’

در ادامه خواهیم دید که چگونه اطلاعات را ذخیره و بازیابی کرد.

Axios یک منبعی برای ذخیره اطلاعات به صورت انلاین می‌باشد که با استفادهاز دستور post اطلاعات را به سمت سرور آن فرستاده و ذخیره خواهیم کرد.

برای اتصال به axios باید یک لینک برای آن ایجاد کرد و در آن آدرس محلی که قرار است داده ها ذخیره شوند را بنویسیم.

در کد زیر خواهیم دید که این اتفاق چطور خواهد افتاد:

axios.post('https://jsonplaceholder.typicode.com/posts/',this.state.list).then(response=>{console.log(response);})

همانطور که در کد بالا مشاهده می‌کنید با استفاده از متد post اطلاعات را به سمت سرور ارسال می‌کنیم.اما این اطلاعات در کجا ذخیره می‌شوند؟

این اطلاعات به صورت object با اطلاعات کلید، محتوا،و سایر موارد در آدرسی که به عنوان آرگومان اول داده شده قرار می‌گیرد.

آرگومان دوم بیان دارد که چه اطلاعاتی باید در این مسیر قرار گیرد و این اطلاعات را با محتوایthis.state.list ذخیره می‌کنیم.

سپس برای آگاهی از صحت اطلاعات که آیا به درستی ذخیره شده است یا خیر با استفاده از متغیر response آنرا در console مرورگر چاپ خواهیم کرد.

برای دیدن این اطلاعات، در صفحه مرورگر با زدن کلیدF12 و رفتن به منوی console قابل مشاهده خواهد بود.

در ادامه لازم داریم اطلاعات ذخیره شده در منع مجازی مان را بازیابی کرده و در داخل پروژه قرار بدهیم.

بنابراین از دستور زیر استفاده کرده تا اطلاعات بازخوانی شوند:

const posts = this.state.posts.map(post => {

return <ListDo id={post.key} content={post.content} checked={post.checked} />

});

بدین صورت اطلاعات را در داخل متغیر ریخته و در آن به جستجو می‌پردازیم.این اطلاعات را در داخل کامپوننت ListDo ذخیره و در آنجا بازیابی می‌کنیم.سپس آن را در بین کدهای html قرار می‌دهیم.

{posts}

* 1. نتیجه‌گیری

همانطور که مشاهده کردید در این فصل به بیان نحوه قرار گیری اجزاء پروژه پرداختیم و بیان کردیم چطور یک وب اپلیکیشن کار می‌کند.

به همین منظور ابتدا یک ساعت با یکسری محتوا طراحی نموده و سپس از فرم ها برای گرفتن اطلاعات استفاده نموده‌ایم.

بعد از صحت اطلاعات در فرم ها، وارد بخش اصلی می‌شویم.

در ادامه بیان کردیم که چطور اطلاعات را بر روی منابع قرار می‌دهیم.با استفاده از حافظه های مرورگر اطلاعات مورد نیاز که قرار است بر روی سایت باقی بماند را از طریق localstorage ذخیره کردیم.

همچنین با استفاده از axios و متدهای آن اطلاعات را بر روی یک سرور مجازی ذخیره و اطلاعات آنرا بازیابی کردیم.

بعد از اتمام این فصل و در ادامه فصل بعدی نحوه کدنویسی و محل برقراری دستورات این فصل را بیان خواهیم کرد.

همچنین در فصل بعدی توضیحات بیشتری درباره ساختار یک پروژه وب اپلیکیشن از جمله کامپوننت‌ها و دستورات آن، بعلاوه محیط کدنویسی خواهیم گفت.

1. پیاده‌سازی
   1. مقدمه

در این فصل قصد داریم درباره جزئیات پرويه بیشتر صحبت کنیم؛ اینکه پروژه چطور نوشته شده است و چگونه کار می‌کند.

از چه دستورات و چه ساختار و برنامه نویسی پیروی می‌کند.

همچنین به این موضوع خواهیم پرداخت که برای نوشتن چنین پروژه از چه ابزارها، نرم افزارها استفاده کرده‌ایم.

در این فصل، درباره چگونگی ایجاد یک وب اپلیکیشن و مفاهیمی که معنای آنرا بیان می‌کند صحبت خواهیم کرد.

* 1. کدنویسی

از دیرباز افراد برای به اشتراک گذاشتن فایل های خود از اینترنت استفاده نموده‌اند که با پیشرفت پی‌در‌پی تکنولوژی این اشتراک گذاری هم پیشرفت چشمگیری داشته است.

کاربران از صفحات سایت برای ارائه گزارش و مطالب گاهی عکس و ویدئو استفاده فراوانی داشته‌اند؛

این اطلاعات اغلب به صورت فایل‌های html می‌گرفتند و کم‌کم با استفاده از css به آنها تنوع می‌دادند.

بعد از مدت‌ها،کاربران تصمیم گرفتنداین صفحات را که فقط به صورت خواندنی بود به صفحات پویا تبدیل کنند.

از همین رو، زبان برنامه نویسی جاوااسکریپت در سال 1997 ابداع شد تا جلوه دیگری به سایت‌ها دهد.

زبان جاوااسکریپت، رفته رفته محبوب شد و برنامه نویسان مرتبا برای آن ویژگی‌های جدیدی ابداع می‌کردند و در سال 2009 نسخه نوینی از این زبان ارائه شد.

این نسخه را ES5[[22]](#footnote-22) نامیدند و در این نسخه تحولی در زبان جاوااسکریپت ایجاد کردند که دستورات را راحت تر و ساده تر انجام می‌داد.

این نسخه از JSON پشتیبانی می‌کرد، آرایه ها اضافه شدند و تغییرات دیگیری نیز صورت گرفت.

در سال 2015 نسخه جدید تری با نام ES6 ارائه شد که علاوه بر ویژگی‌های ES5 امکان تعریف متغیر let و ثابت const را بوجود آورد.

با گذشت زمان و محبوب شدن نرم افزارهای موبایلی و شبکه های اجتماعی، افراد کمتری از سایت‌ها استفاده می‌نمودند.

از طرفی، این نرم افزار‌ها گاهی با ورژن‌های خاص کار می‌کردند و با قدیمی شدن نسخه های موبایلی قابلیت اجرا نداشتند.

بنابراین، نرم افزارهایی نوشته شدند که هم توسط نرم افزارها و هم از طریق مرورگرها قابل اجرا باشند.

به این شکل وب اپلیکیشن‌ها به وجود آمدند.

وب اپلیکیشن‌ها، همان خدمات نرم افزارهای موبایلی را اما باکمی تفاوت بر روی مرورگر ارائه می‌دادند.

محبوب ترین این وب اپلیکیشن ها فیس بوک[[23]](#footnote-23)و جی میل [[24]](#footnote-24)هستند.

این وب اپلیکیشن ها نیازمند برنامه نویسی ظاهر سایت و پشت صحنه سایت است، که پشت صحنه آن را با زبان های برنامه نویسی php ،C# و... می‌نویسند و ظاهر سایت را با زبان جاوااسکریپت.

برنامه نویسان جاوااسکریپت کتابخانه ها و چارچوب هایی برای استفاده ساده تر و جذاب تر آن با عنوان های مختلفی ارائه کردند که یکی از آن کتابخانه ها React نام دارد.

این کتابخانه اولین بار در سال 2011 توسط شرکت facebook منتشر شد و در سال 2013 به طور رسمی مورد استفاده قرار گرفت.

این کتابخانه، ظاهری بر منای ui برای کدهای جاواسکریپت ایجاد می‌کند.با استفاده از این کتابخانه، بجای تغییر مستقیم DOM، یک DOM مجازی ایجاد می‌کند و تغییرات را بر روی آن اعمال می‌کند.سپس اگر نیاز به تغییر بود آنرا دستکاری کرده و در انتها آن را در DOM اصلی اعمال می‌کند.

آخرین نسخه آن V5 است که در سال 2022 منتشر شد.

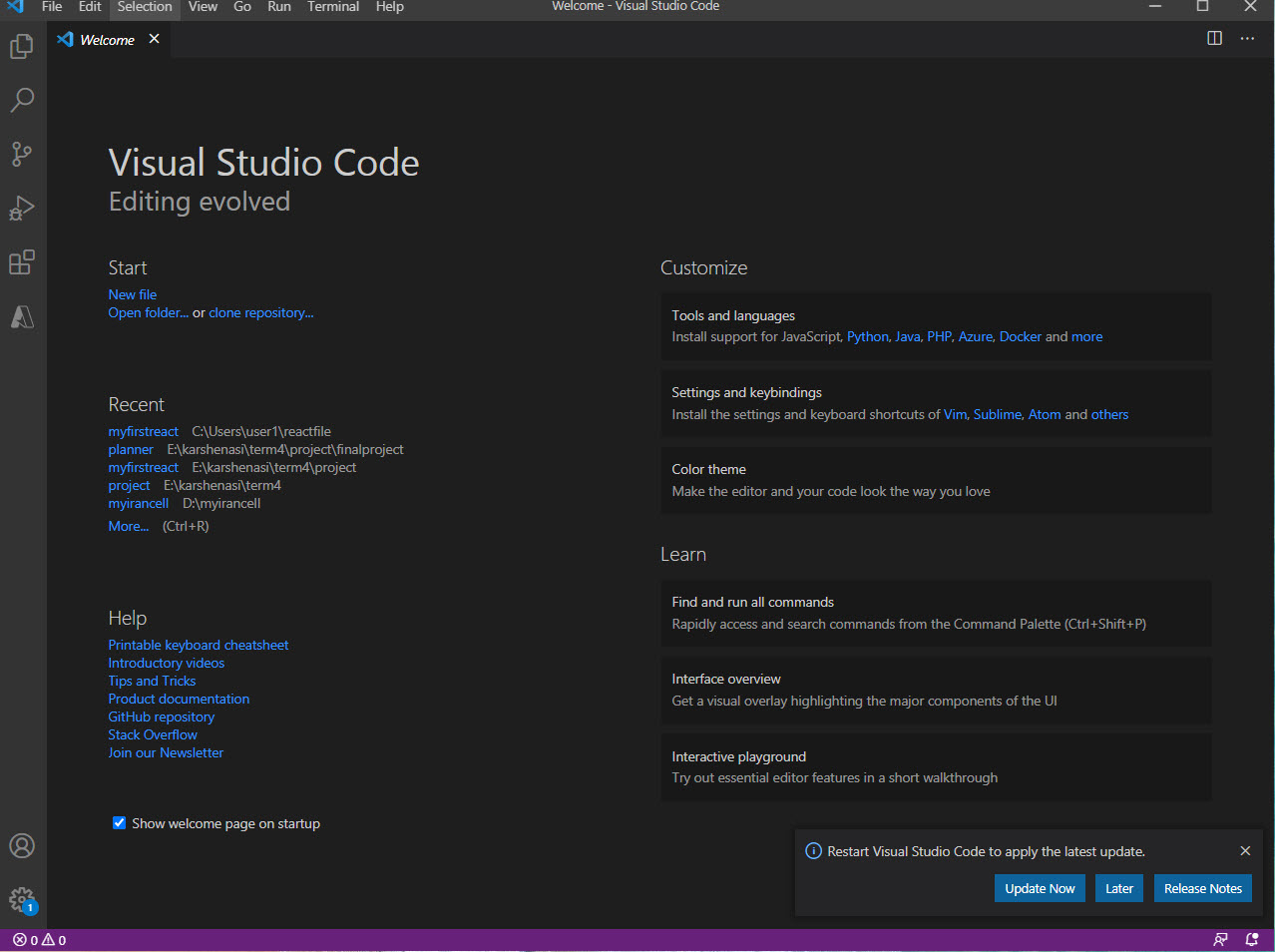
برای استفاده از کتابخانه React نیاز به پلتفرمی داریم تا آنرا برای محیط برنامه نویسی ما شناسایی کند و به عنوان زبان برنامه نویسی سمت backend سایت پشتیبانی کند.

از این رو از پلتفرم Nodejs [[25]](#footnote-25)استفاده می‌کنیم که زبان جاوااسکریپت را به خوبی شناسایی می‌کند.

Node.js از نظر طراحی شبیه به سیستم هایی مانند Ruby's Event Machine و Python's Twisted است و تحت تأثیر آنها قرار گرفته است. Node.js مدل رویداد را کمی جلوتر می برد. این یک حلقه رویداد را به‌عنوان ساختار زمان اجرا به جای کتابخانه ارائه می‌کند. در سیستم های دیگر، همیشه یک تماس مسدود کننده برای شروع حلقه رویداد وجود دارد. به طور معمول، رفتار از طریق callbacks در ابتدای یک اسکریپت تعریف می‌شود و در پایان یک سرور از طریق یک تماس مسدودکننده مانند EventMachine::run () شروع می‌شود. در Node.js، چنین فراخوانی حلقه شروع رویداد وجود ندارد. Node.js به سادگی پس از اجرای اسکریپت ورودی وارد حلقه رویداد می شود. Node.js زمانی از حلقه رویداد خارج می شود که دیگر تماسی برای انجام وجود نداشته باشد. این رفتار مانند جاوا اسکریپت مرورگر است - حلقه رویداد از کاربر پنهان است.

به علاوه برای کد نویسی و استفاده از کتابخانه React نیاز به محیط برنامه نویسی با عنوان Visual Studio Code که به اختصار VSCode [[26]](#footnote-26)نام برده می‌شود.قابل ذکر است که این محیط با Visual Studio کاملا متفاوت است و برخلاف آن کار کردن با آن بسیار راحت بوده و برای نصب نیاز به فریم ورک [[27]](#footnote-27)خاصی ندارد.

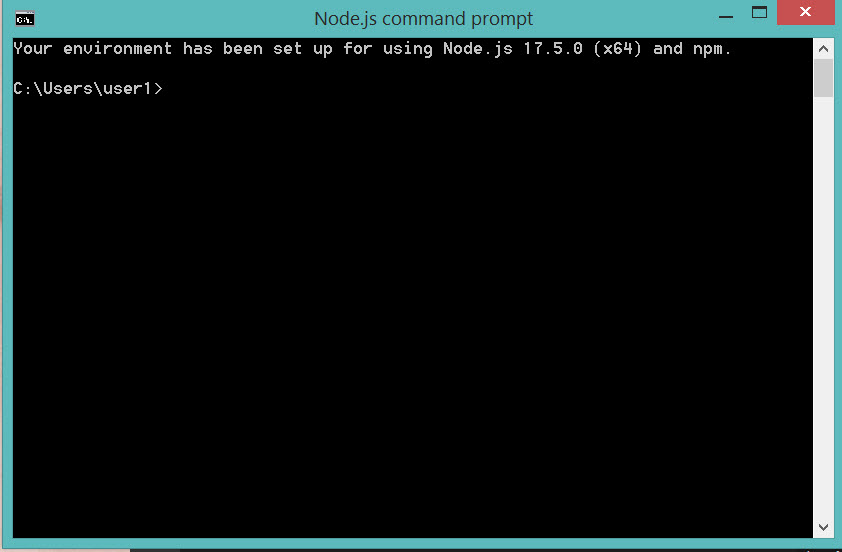
تصویر4-1 محیط برنامه نویسی vs code



برای شروع برنامه نویسی با کتابخانه react ابتدا باید در nodjs یک پروژه در این باره ایجاد کنیم.

به این منظور به nodejs command propmt رفته یا با نصب git bash[[28]](#footnote-28) این کار را انجام خواهیم داد.

تصویر4-2 محیط خط فزمان nodejs



ویژگی Git که واقعاً آن را از تقریباً هر SCM دیگری که وجود دارد متمایز می کند، مدل شاخه ای آن است.

Git به شما اجازه می دهد و شما را تشویق می کند که چندین شعبه محلی داشته باشید که می توانند کاملاً مستقل از یکدیگر باشند. ایجاد، ادغام و حذف این خطوط توسعه چند ثانیه طول می کشد.

این بدان معنی است که شما می توانید کارهایی مانند:

تغییر متن بدون اصطکاک. یک شاخه ایجاد کنید تا یک ایده را امتحان کنید، چند بار متعهد شوید، به جایی که از آن منشعب شده اید برگردید، یک وصله اعمال کنید، به جایی که در حال آزمایش هستید برگردید و آن را با هم ادغام کنید.

کدهای مبتنی بر نقش. شعبه‌ای داشته باشید که همیشه فقط شامل مواردی باشد که به تولید می‌رسد، شاخه دیگری که کار را برای آزمایش در آن ادغام می‌کنید و چندین شاخه کوچکتر برای کار روزانه داشته باشید.

گردش کار مبتنی بر ویژگی. برای هر ویژگی جدیدی که روی آن کار می‌کنید، شاخه‌های جدیدی ایجاد کنید تا بتوانید به‌طور یکپارچه بین آن‌ها جابه‌جا شوید، سپس وقتی آن ویژگی در خط اصلی شما ادغام شد، هر شاخه را حذف کنید.

آزمایش یکبار مصرف یک شعبه برای آزمایش ایجاد کنید، متوجه شوید که کار نمی کند، و فقط آن را حذف کنید - کار را رها کنید - بدون اینکه هیچ کس دیگری هرگز آن را نبیند (حتی اگر شاخه های دیگر را در این بین فشار داده باشید).



تصویر4-3محیط خط فرمان git bash

برای ایجاد پروژه وارد قسمت خط فرمان شده و دستور زیر را می‌نویسیم:

Npm install –g create-react-app

پس از نصب شدن یک پوشه با نام myfirstreact ایجاد می کنیم.قابل ذکر است پوشه ای که ایجاد کرده ایم نام پروژه ما نیز خواهد بود:

Npx create-react-app myfirstreact

سپس به داخل پوشه تغییر مسیر داده:

Cd myfirstreact

و آن را اجرا می‌کنیم:

Npm start

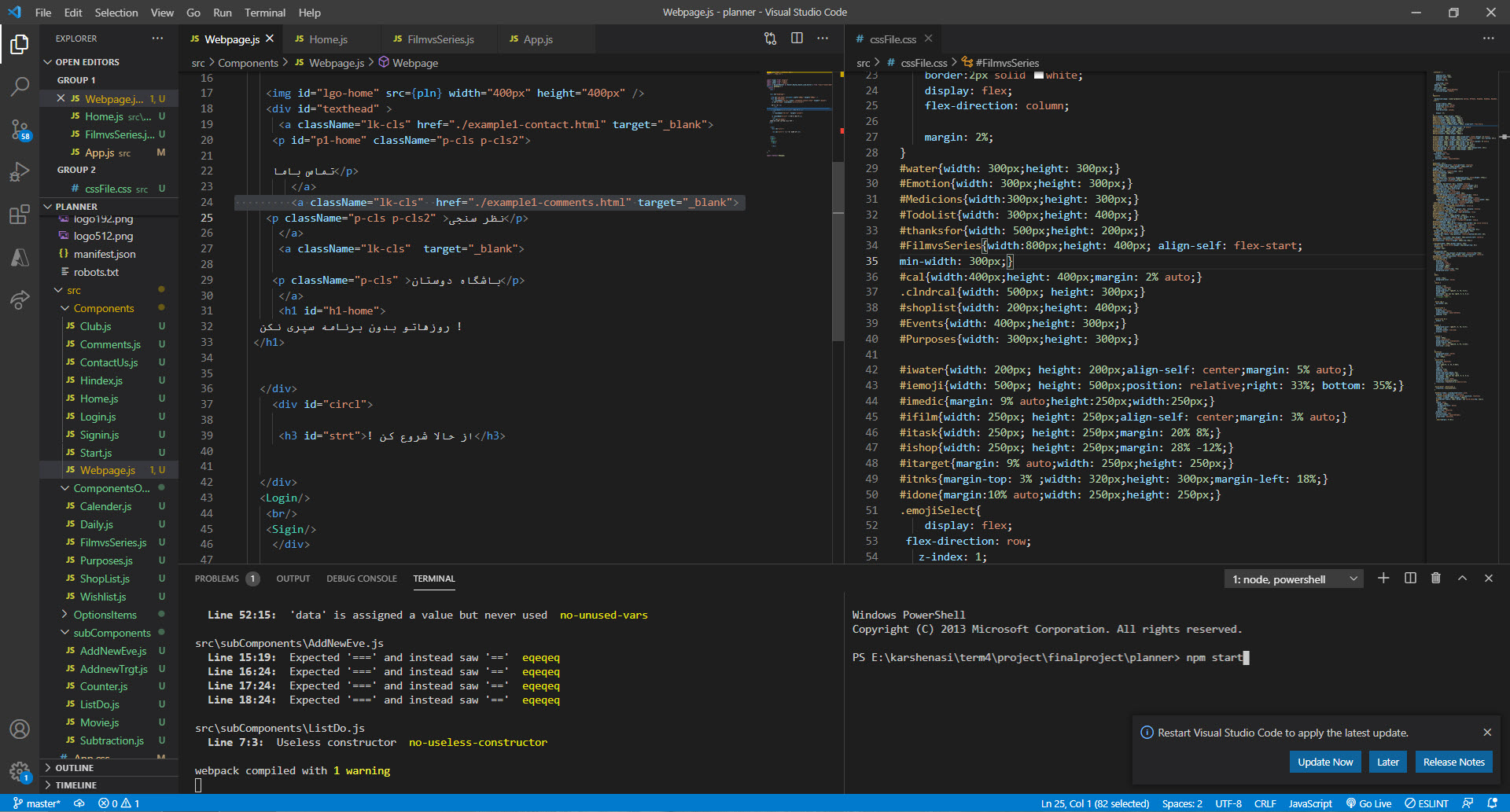
به طور خودکار با برنامه vs code پروژه ما اجرا خواهد شد.

اما اگر بخواهید به روش دیگری اجرا کنید، پس از ساخت پروژه، در داخل پوشه راست کلیلک و بر روی git bash here کلیک کنید.

سپس . code را نوشته و enter بزنید.

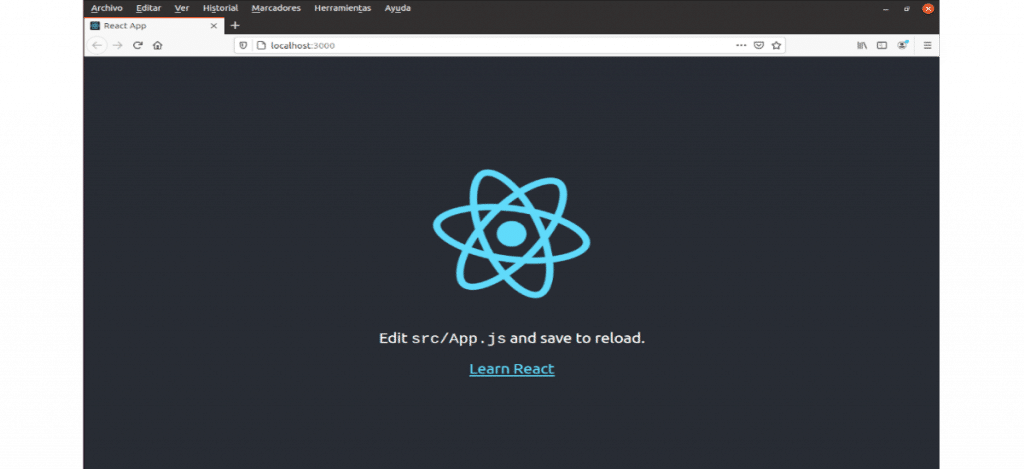
در داخل برنامه vs code که باز می‌شود، در قسمت ترمینال، عبارت npm start را وارد نمایید.

تصویر4-4 اجرای پروژه



به این صورت اولین پروژه ای که ساختیم در صفحه مرورگر که به طور پیش فرض chrome است اجرا می‌شود.این پروژه بر روی پورت 3000 به صورت لوکال هاست قابل مشاهده است.

تصویر 4-5 اجرای اولین پروژه بر روی مرورگر



سپس به پروژه برگشته و کدهای آنرا پاک می‌کنیم و کدهای مربوطه را می‌نویسیم.

برای نوشتن وب اپلیکیشن مورد نظر، به این صورت در نظر گرفته‌ایم که ابتدا یک وب سایت همانند سایر سایت‌ها داشته باشیم.سپس در این سایت یکسری اطلاعات به همراه فرم ها در نظر گرفته ایم که کاربران با احراز هویت و ثبت نام بتوانند صفحه‌ای شخصی در وب اپلیکیشن داشته باشند.

از این رو یک کامپوننت [[29]](#footnote-29)به نام webpage در کامپوننت App ایجاد می‌کنیم.

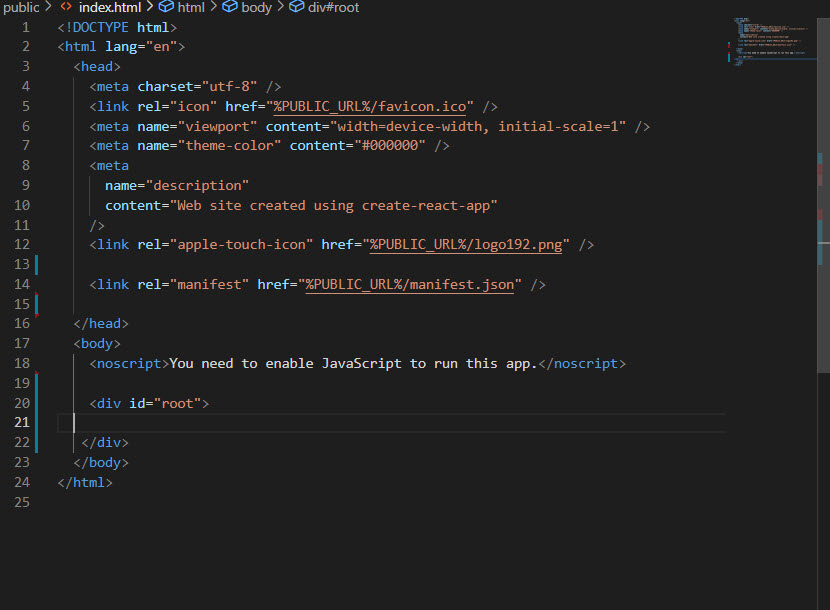
کامپوننت ها،ساختار اصلی یک پروژه ری اکت را بیان می‌کنند.

اگر به صفحه محیط vs code نگاه کنید، خواهید دید یک فایل inedx.html یک فایل index.js و یک فایل index.css داریم.همانطور که میدانید index.css استایل دهی صفحات html هستند.

در فایل index.html یک سری کد html داریم که در آن فقط یک div با شناسه[[30]](#footnote-30)root قرار داده شده است.

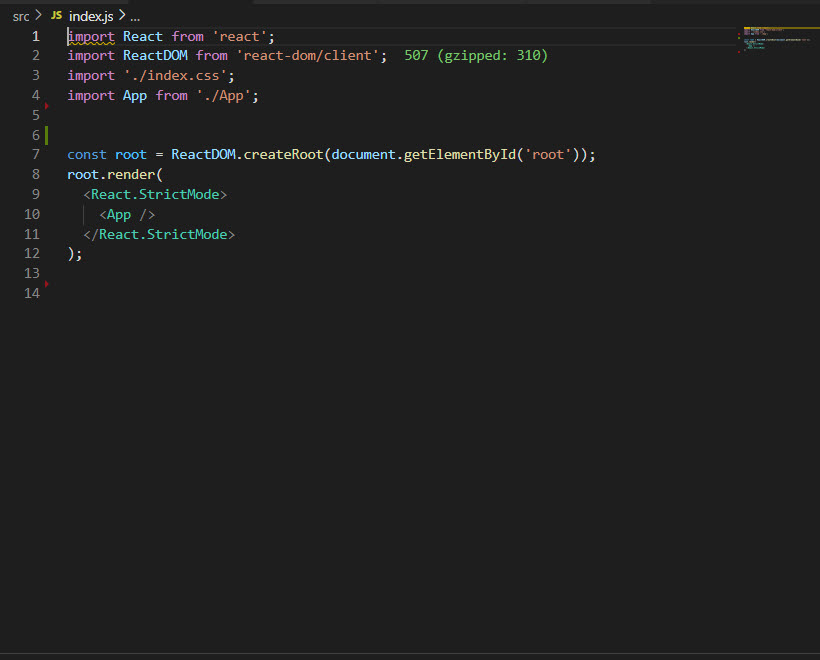
به این علت که صفحات html پروژه ما با کتابخانه ری اکت به صورت single page application قرار است نوشته شود.به این معنا که تمامی کدهایی که قرار است نوشته شود فقط در داخل یک المنت قرار می‌گیرد.

تصویر4-6 المان با شناسهroot



در فایل index.js که فایلی با فرمت جاوااسکریپت است مشخص می‌کنیم چه کدهایی را در این المان در فایل index.html اجرا کند.

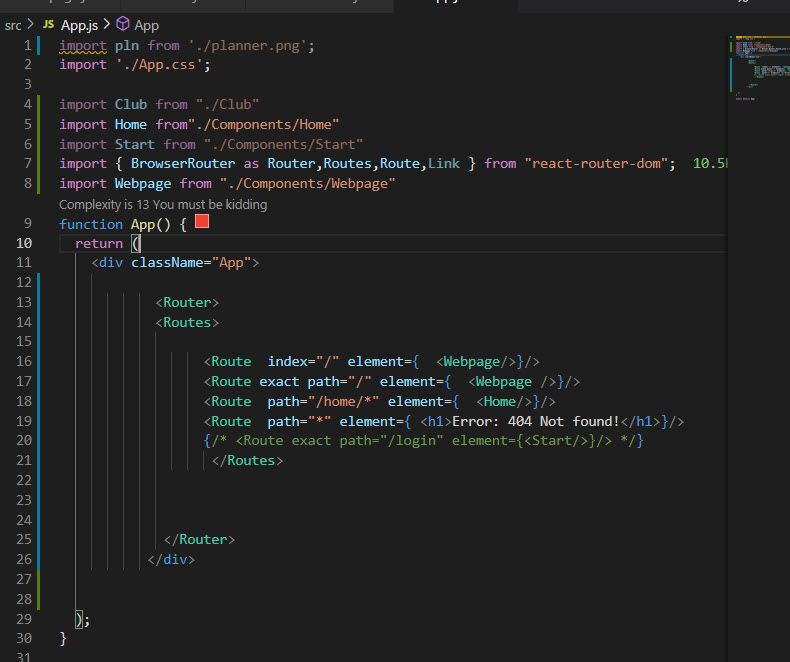
تصویر4-7 کدهای فایل index.js



در تصویر بالا تگی با عنوان App داریم.این تگ در واقع یک کامپوننت است که کدهای مورد نظرمان را در داخل آن قرار داده ایم.

در کامپوننت App، دو کامپوننت دیگر قرار داده ایم که یکی برای صفحه اصلی و دیگری برای صفحه وب اپلیکیشن ما هست.

تصویر4-8 کامپوننت App

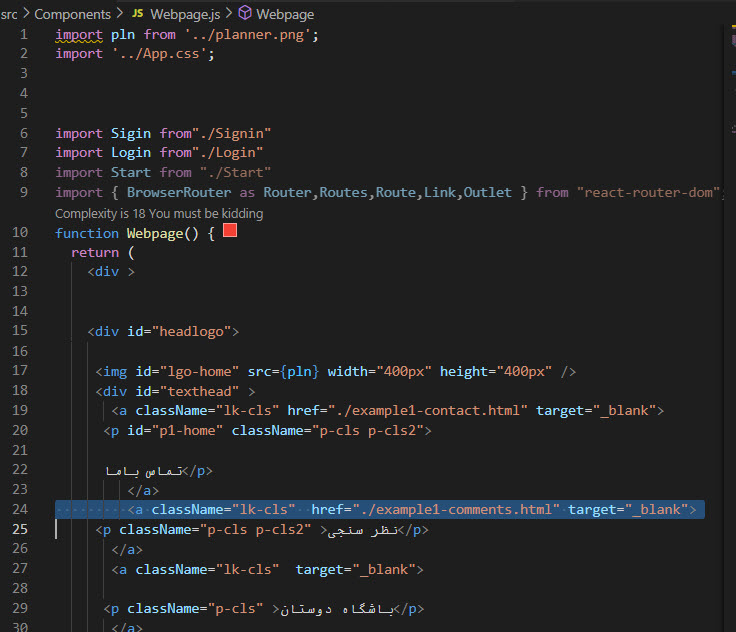


همانطور که در تصویر بالا مشاهده می‌فرمایید،برای ایجاد دسترسی به صفحات از Router استفاده کرده و آنهارا آدرس دهی کرده‌ایم.

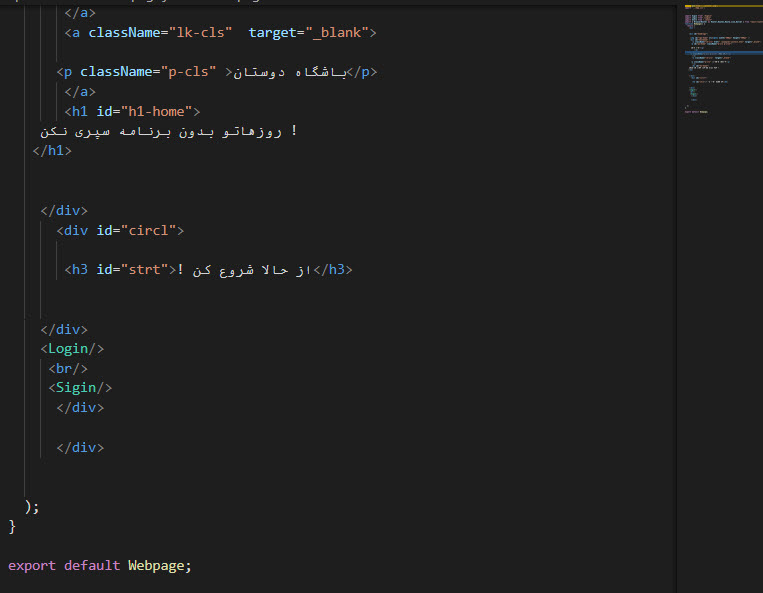
صفحه اصلی را webpage و برای هرچیز نامربوطی که جستجو شود 404 Error را در نظر گرفته ایم.

کامپوننت Webpage صفحه اصلی و کامپوننت Home صفحه وب اپلیکیشن ما هست.

تصویر4-9 صفحه وب پیج



تصویر4-10 صفحه وب پیچ



در کامپوننت وب پیج لینک ها، فرم ها،تصویر لوگو و اطلاعاتی در مورد سایت ارائه شده است.

در این کامپوننت کدها به صورت JSX[[31]](#footnote-31) نوشته شده است.به این معنا که کدهای html را در کدهای جاوااسکریپتی به کار برده‌ایم.

در این صفحه از دو کامپوننت دیگر برای بخش ثبت نام و ورود استفاده کرده‌ایم.

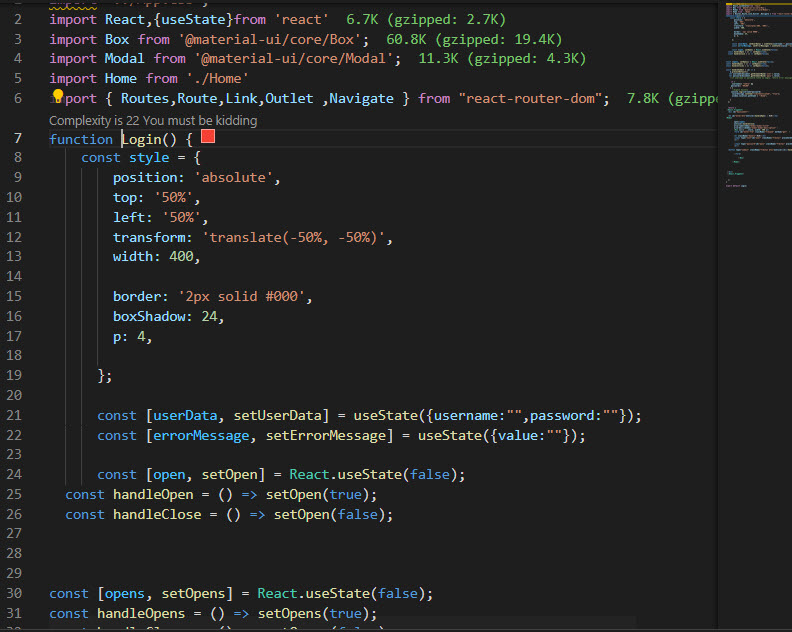
این دو کامپوننت را به صورت مدال[[32]](#footnote-32) نوشته ایم.

به اسن صورت که کاربر زمانی که بر روی ورود یا ثبت نام کلیک می‌کند صفحه ای بر روی صفحه سایت باز می‌شود و با کلیک بر اطراف می‌توان بر صفحه اصلی سایت به راحتی بازگشت.

بدین صورت، برای احراز هویت دیگر نیاز به مراجعه به صفحه‌ای جداگانه نیست.

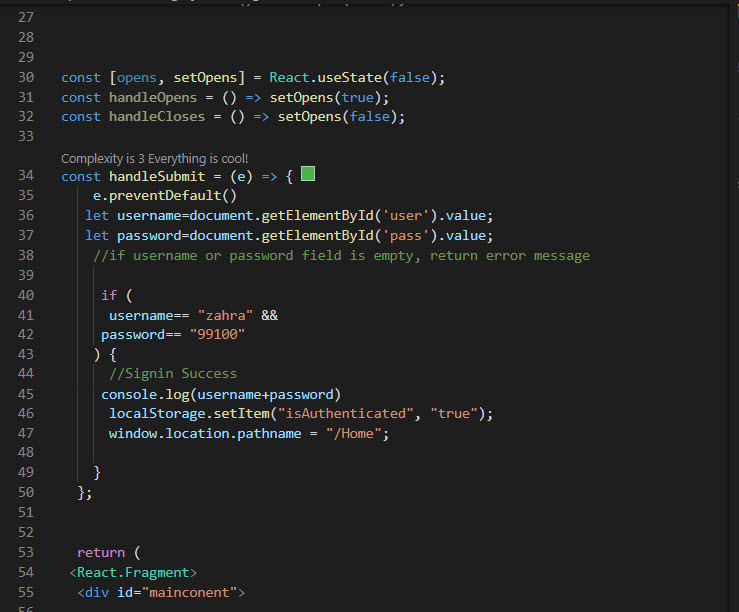
لازم به ذکر است، برخی کامپوننت ها بعلت ضرورت به صورت Functional و برخی دیگر به صورتClass نوشته شده است.

از مزایای کتابخانه ری‌اکت،پشتیبانی کردن از شیءگرایی است.این بدین معناست که می‌توان شیءهایی ایجاد کرد که از خاصیت ارث‌بری پشتیبانی کنند.همچنین،برای نوشتن کامپوننت های تابعی می‌توان از arrow function هم استفاده نمود.



تصویر4-11 کامپوننت ورود

تصویر4-12 کامپوننت ورود



تصویر4-13 کامپوننت ورود



برای نوشتن صفحات مدال، نیازمند به استفاده از کتابخانه هایی هستیم که توسط ری اکت پشتیبانی شوند.بنابراین از توسعه دهنده material ui کتابخانه modal را بر روی پروژه با دستور زیر نصب می‌کنیم:

Npm install material –ui modal

پس از نصب، با مراجه به فایل Package.json می‌توان اطلاعاتی مربوط به کل پروژه و فایل های نصب شده اعم از نسخه نصب شده را در این فایل مشاهده کرد.

در ادامه باید این کتابخانه را در پروژه خود شناسایی کنیم تا بتوانیم از آن استفاده کنیم.

پس با استفاده از دستور زیر آنرا به فایل مورد نظر اضافه می‌کنیم:

Import Modal from ‘@materialui/core/modal’

در ادامه برای نوشتن صفحات مدال نیاز به استفاده از hook ها داریم.hook ها کتابخانه های داخلی ری‌اکت هستند که بدون نوشتن توابع و کلاس ها می‌توان از قابلیت های ری اکت استفاده نمود.

بنابراین، آنرا از کتابخانه ری اکت با دستور زیر فراخوانی می‌نماییم:

Import React, {useState} from ‘react’

Hook مورد استفاده ما در این فایل usestate است.دقت شود که باید این کلمه به صورت camelcase نگاشته شود.

ازusestate در این قسمت از پروژه، برای متغیرهای component ها احتیاج داریم و به صورت زیر آنها را در برنامه تعریف می‌کنیم.

در ادامه خواهیم دید چطور از آنها در برنامه استفاده کنیم:

const [userData, setUserData] = useState({username:"",password:""});

در این خط کد،یک متغیر از نوع const با نام userData تعریف کرده ایم و قرار است هرجا نیاز شد با setUserData مقدار آنرا قرار دهیم.

این متغیر، از نوع شیء است و دومقدار username و password قرار است در آن قرار گیرد.

در ادامه، زمانی که بخواهیم مقدار userData را قرار دهیم با استفاده از setUserData این کار را انجام می‌دهیم؛ در نتیجه username و password مقدار دهی می‌شوند.

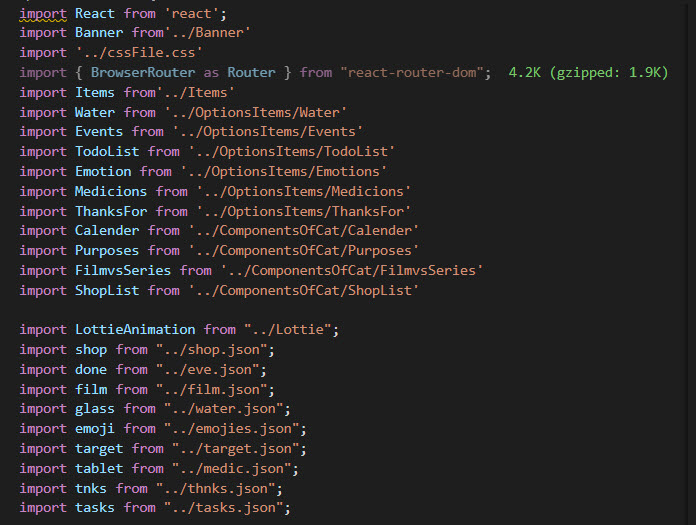
در ادامه، تابع handleSubmit را داریم که با کلیک بر روی دکمه ورود بررسی می‌کند که آیا فیلدهای نام کاربری و رمز عبور صحیح می‌باشد یا خیر.

در صورت صحیح بودن،احراز هویت آن تایید می‌شود و وارد صفحه Home می‌شود.

پس از هدایت شدن به صفحه Home آدرس مرورگر تغییر کرده و کامپوننت Home اجرا می‌شود.

این کامپوننت، کامپوننت اصلی این پروژه می‌باشد؛زیرا هدف اصلی ما از این پروژه این بخش است و تمامی کدهای اصلی و منظور در این بخش لحاظ گردیده است.

تصویر4-14 کامپوننت Home



همانطور که گفته شد،این کامپوننت حاوی کدهای اصلی پروژه می‌باشد و در این فایل کامپوننت‌های فرعی را فراخوانی کرده ایم. همانطور که در تصویر بالا مشاهده می‌شود کامپوننت های فرعی که مربوط به هر بخش این پروژه می باشد فراخوانی شده اند.

این کامپوننت ها هرکدام اطلاعاتی جدا از هم دارند که در هر بخش یکسری عملیات خاص و منحصر به فرد اتفاق می‌افتد.

به علاوه برای فهم ساده تر این کامپوننت ها برای کاربر(زیرا کاربران ممکن است اطلاعی از کامپوننت نداشته باشند) از انیمیشن ها برای توصیف آنان استفاده نموده ایم.

برای بیان این انیمیشن ها از کتابخانه Lottie استفاده کرده ایم تا برای ما انیمیشن ها را تعریف کند.

همپنین المان های انیمیشنی را دانلود کرده و با پسوندjson ذخیره می‌کنیم.به این صورت ری اکت آن ها را تشخیص می‌دهد.

درواقع فایل های انیمیشن ما به صورت json ذخیره می‌شوند و با Lottie قرار است بگوییم این انیمیشن ها چطور اجرا شوند؛مثلا پس از پایان یافتن تایم اجرا دوباره اجرا شوند.

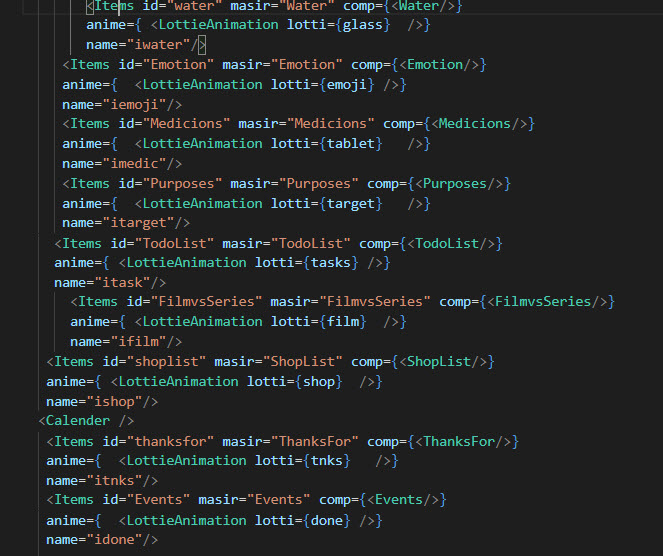
loop: true,

با استفاده از دستور زیر به این کتابخانه خواهیم گفت داده انیمیشنی ما کدام است:

animationData: lotti,

سپس آنرا به کامپوننت Item فرستادیم تا در حجم کدنویسی و ایجاد کامپوننت ها صرفه جویی کنیم.

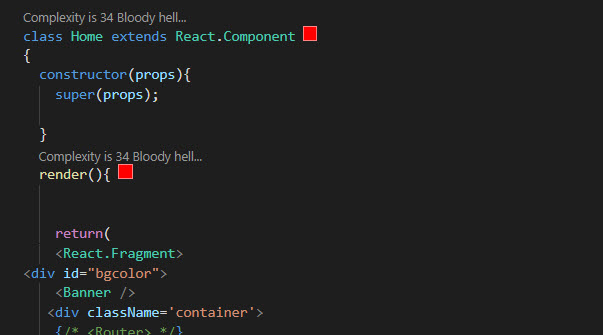
برای ایجاد کامپوننت های مختلف با مفهوم یکسان اما کاربرد متفاوت، کامپوننت Item را ایجاد کرده ایم.دراین کامپوننت id و یکسری اطلاعات لازم از جمله نام، مسیر و کامپوننتی را که قرار است اجرا کند را فرستاده‌ایم.این کار را با استفاده از property ها ارسال کررده و در کامپوننت Item با مفهوم Props آنرا دریافت کرده ایم.



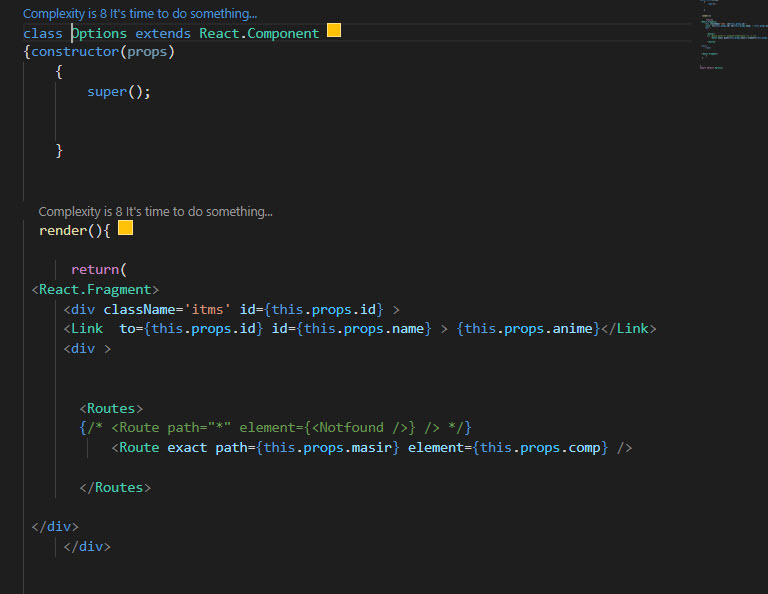
تصویر4-15 کامپوننت Item

برای شخصی سازی این وب اپلیکیشن از کامپوننت دیگری به نام Banner استفاده کرده‌ایم که اطلاعات شخصی فردی را که لاگین می‌کند را نمایش دهیم.

تصویر4-16 کامپوننت Banner



تصویر4-17 کامپوننت Items



در این کامپوننت اطلاعات را با استفاده از props دریافت کرده و نمایش می‌دهیم.

سپس برای هر کدام link و route در نظر می‌گیریم تا بتوانند هرکدام زیر مجموعه خود را نشان دهند.

برای استفاده از مسیریابی و لینک ها باید react router dom را نصب داشته باشید.به همین منظور در ترمینال VsCode دستور زیر را وار می‌کنیم:

Npm I react-router-dom –save

پس از نصب آنرا با دستور زیر به پروژه اضافه نموده و از ویژگی هایی که اراده می‌دهد استفاده می‌کنیم:

import { Link, Route,Routes } from “react-router-do”;

آیتم هایی که در این پروژه قرار داده‌ایم، هر یک از دیگری مستقل هستند و بخش های مختلف پروژه را تشکیل می‌دهند.

آیتم های این پروژه عبارتند از مقدار آب مصرفی، داروهای مصرفی،هدف ها،احساسات شخص،فیلم وسریال،لیست کارها،لیست خرید،تقویم ، چالش شکرگزاری رویدادها.

در ادامه این فصل، با این آیتم ها بیشتر آشنا خواهیم شد اما اکنون قرار است توضیح دهیم هر یک چه کاری انجام می دهند و چگونه عمل می‌کنند.

در این کامپوننت ها از آیکون ها، Input ها،دکمه ها و Api [[33]](#footnote-33)ها استفاده کرده ایم.

در کامپوننت Water،Emotions،Purposes،Medicions از آیکون ها و اینپوت ها استفاده کرده ایم.

که آیکون ها بیانگر هدف از کامپوننت است و توسز اینپوت ها داده مورد نظر را وارد کرده و با دکمه ها آنرا به ثبت می‌رسانیم.

در کامپوننت های Todolist و Shoplist یک لیست از کارهای درخواستی تهیه می‌کنیم.

در چالش شکرگزاری سعی کرده ایم یک راه انگیزشی با کامپوننت ThanksFor ایجاد کنیم.

بعلاوه از api ها کمک گرفتیم تا فیلم ها را به نمایش بگذاریم و همپنین تقویم فارسی ارائه دهیم.

در انتها آنها را در منابع مون ذخیره می‌کنیم.

کدهای مهمی که در این کامپوننت ها استفاده شده است به صورت زیر می‌باشد:

event.preventDefault();

با استفاده از دستور بالا در توابع، از refresh شدن مجدد صفحه جلوگیری خواهیم کرد.

key:UUID(),

با استفاده از کتابخانه UUID می توان شناسه های منحصر به فرد برای ممتغیر ها در نظر گرفت.

let newlist=prevstate.list.map((row)=>{if(row.key===key){row.checked=!row.checked;}return row})

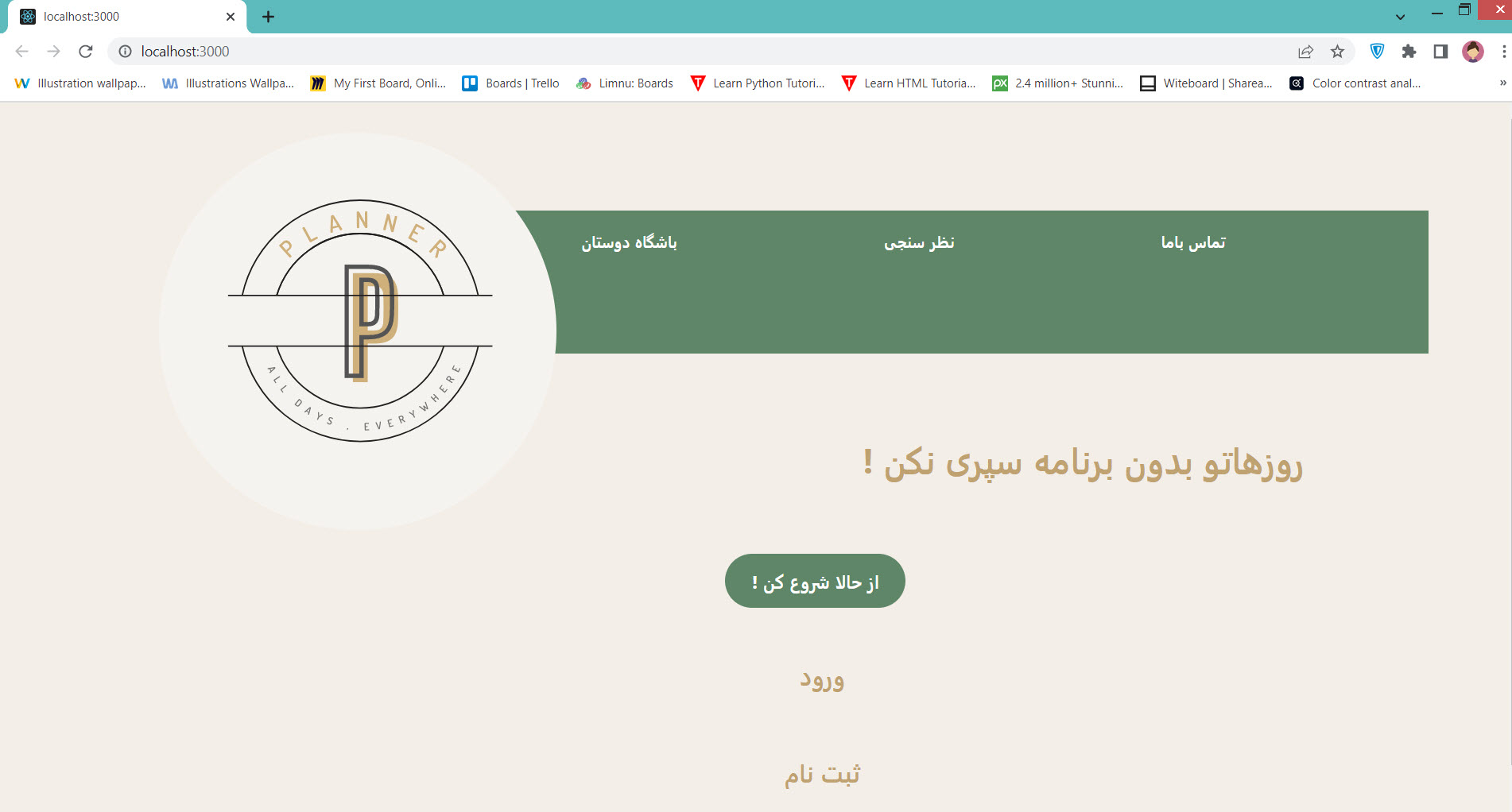
return{list:newlist}

})

در کد بالا، مشاهده می‌شود که با دستور map همانند حلقه for در آرایه جستجو می‌شود و مقادیر آنرا مورد بررسی قرار می‌دهد.

* 1. اجرای نرم­افزار

در ادامه خواهیم دید پس از کد نویسی قسمت های مختلف یک پروژه، به چه نتیجه‌ای خواهیم رسید.

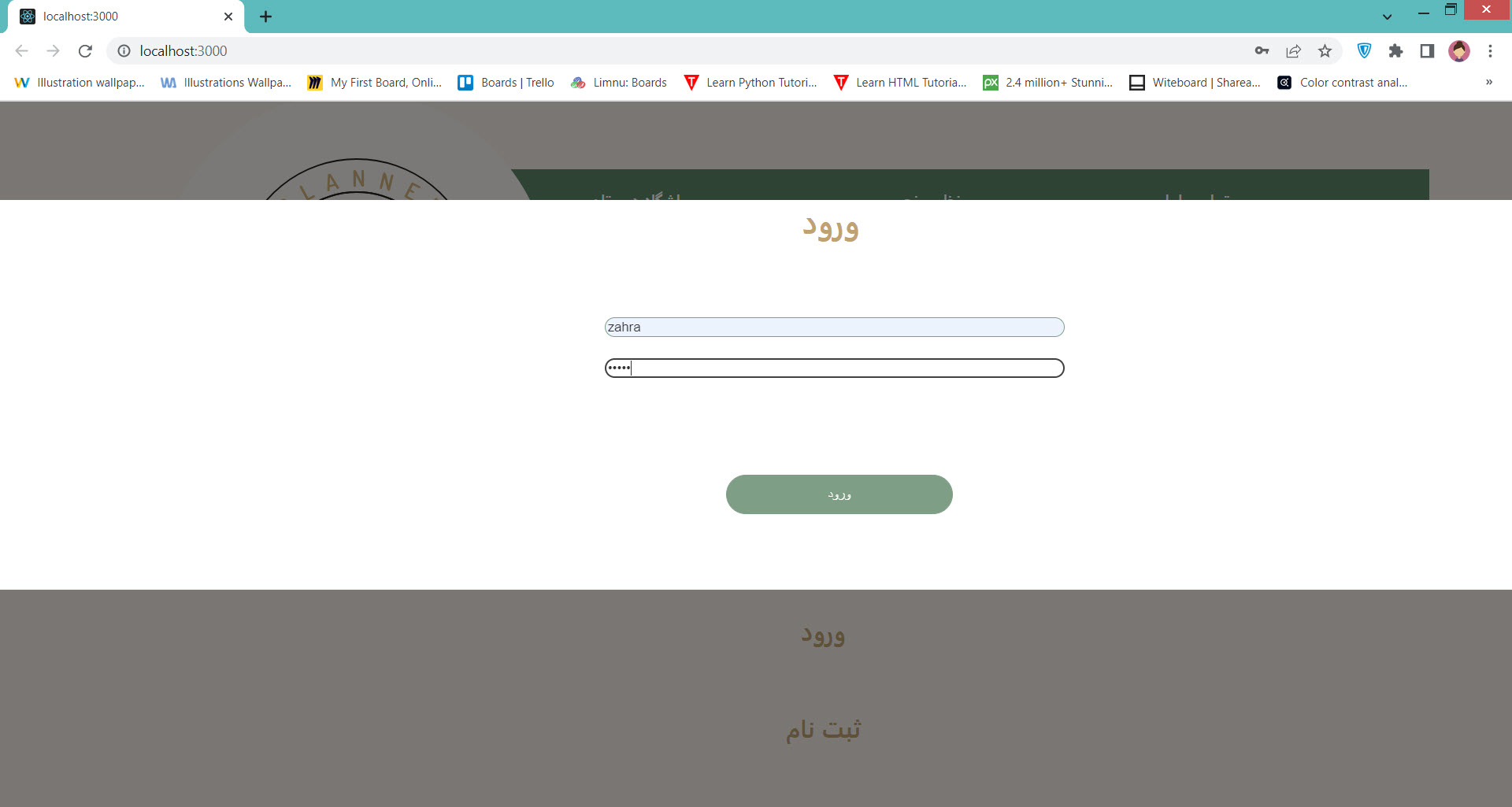


تصویر4-18 صفحه اصلی

در تصویر بالا،پس از بالا آمدن پروژه،اولین چیزی که لود می‌شود این صفحه است.

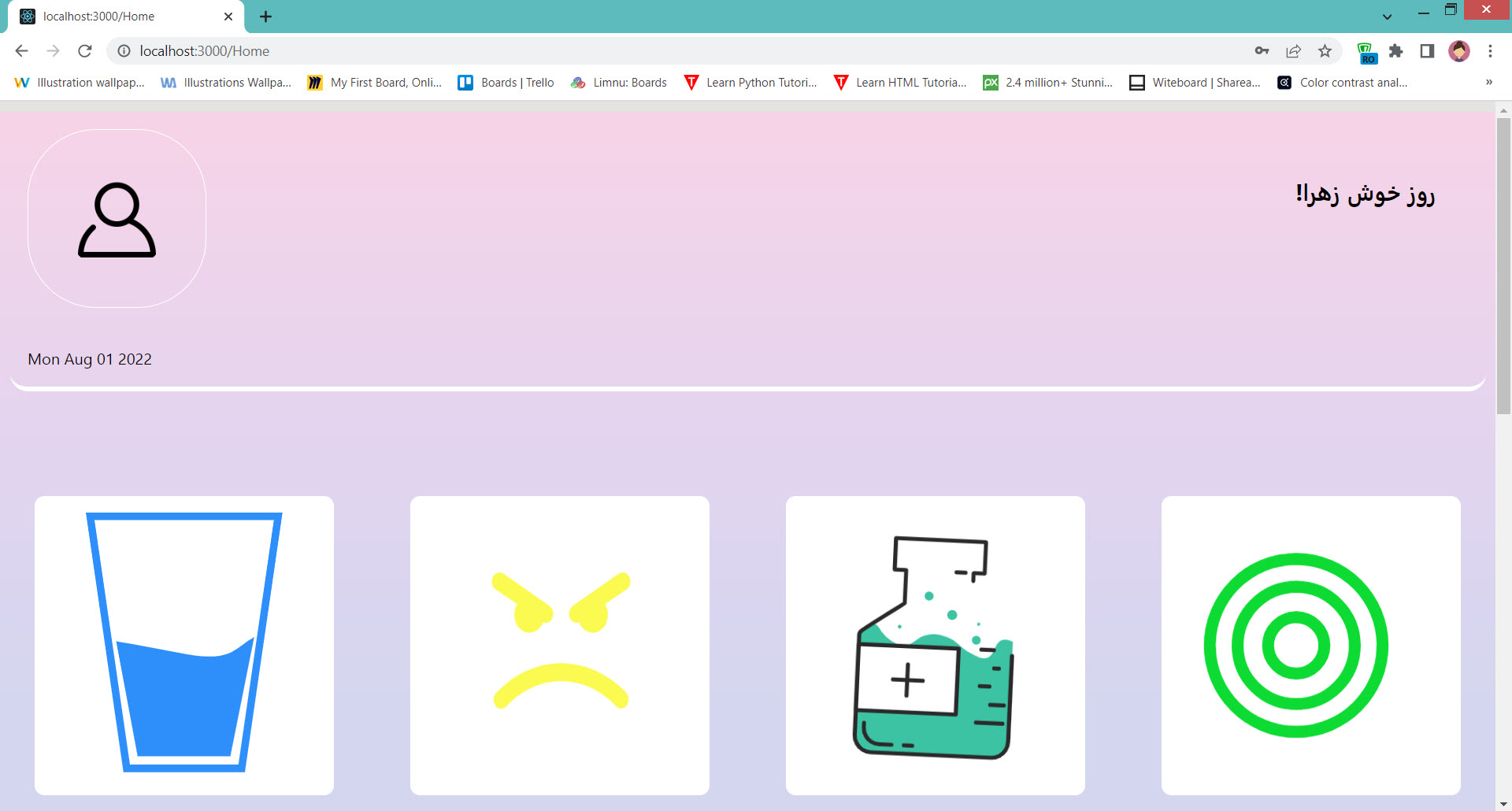
با کلیک بر روی هرکدام از المان ها یک فرم باز می‌شود که در تصویر زیر خواهید دید.

تصویر 4-19 فرم لاگین



همانطور که قبلا گفتیم فرم ها را به صورت مدال نوشته ایم.این فرم دارای دو فیلد ورودی می باشد و با زدن دکمه ورود، در صورت صحیح بودن عبارات وارد صفحه Home می شود و آدرس لینک را تغییر می دهد که این کار را با استفاده از Router انجام داده‌ایم.

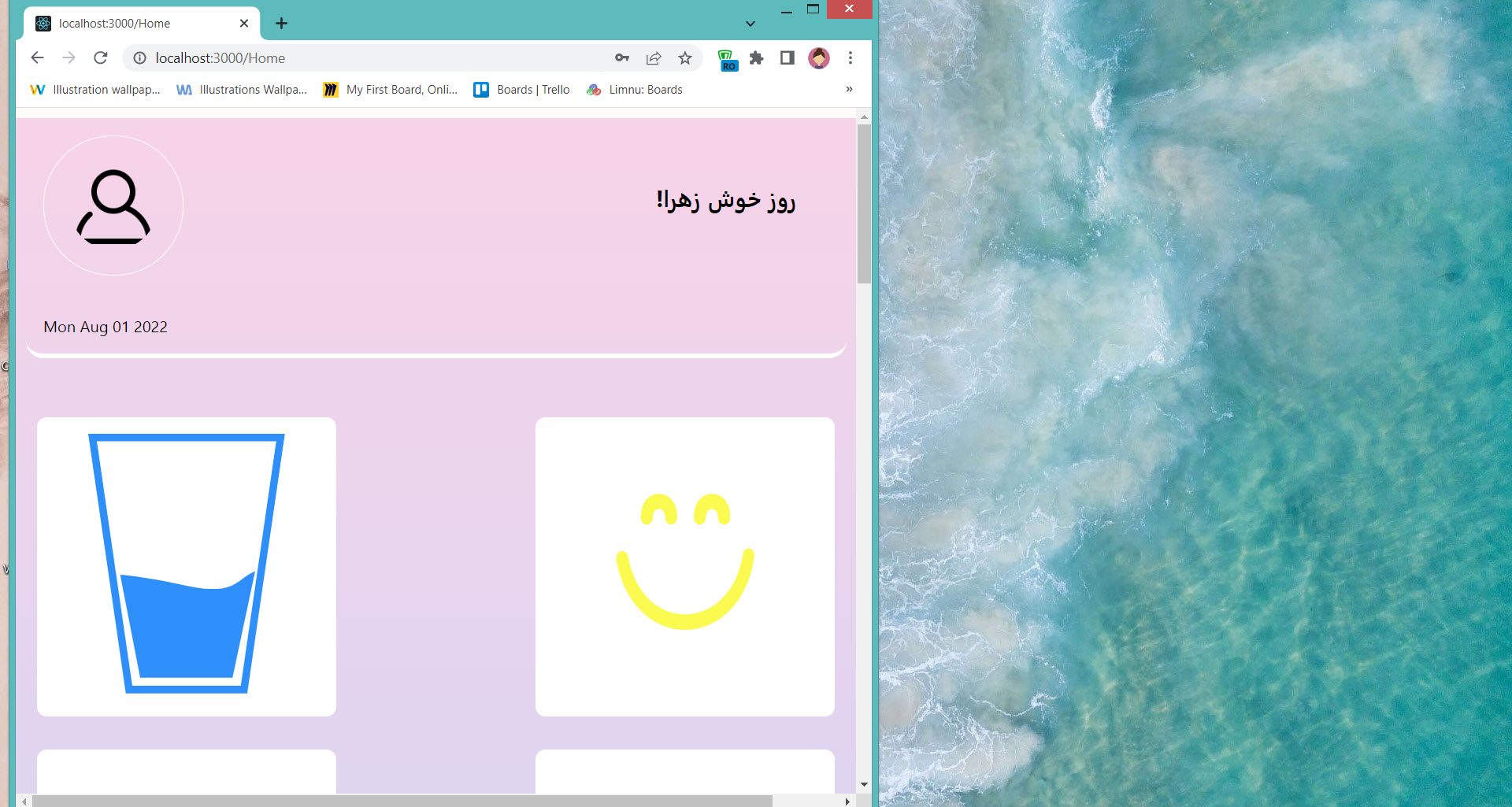
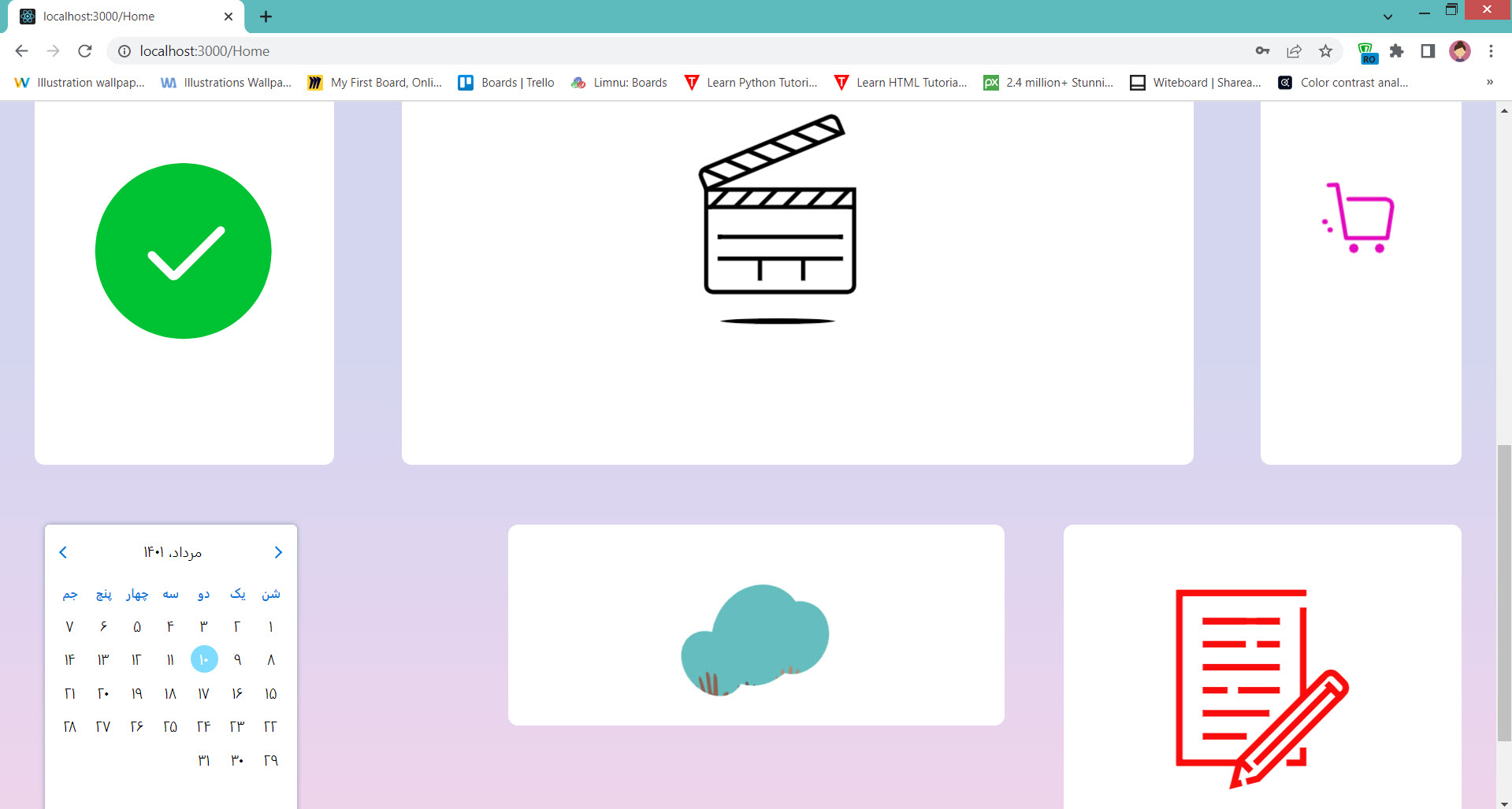
تصویر4-20 بخشی از صفحه Home



در این صفحه همانطور که مشاهده می‌فرمایید آیتم هایی که بیان کرده ایم با مقادیر انیمیشن های طراحی شده تعبیه شده اند.

هر کدام مقصودی را بیان می‌کنند و حاوی زیرکامپوننت ها هستند.

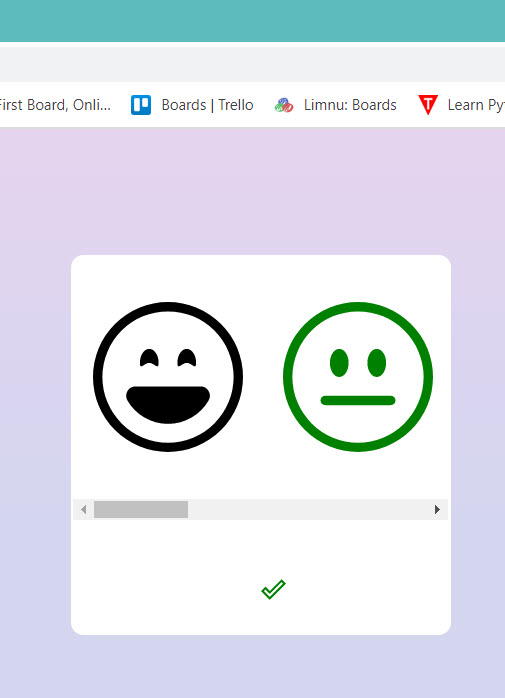
تصویر4-21 ادامه صفحه Home



تصویر4-22 ریسپانسیو بودن سایت

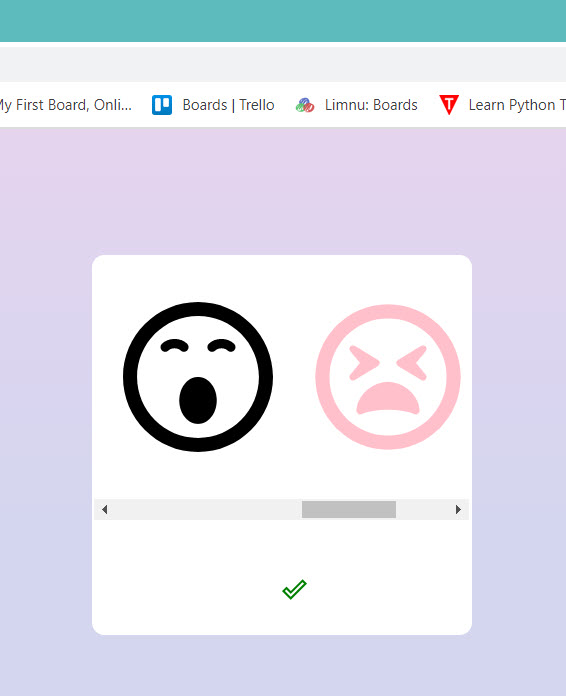
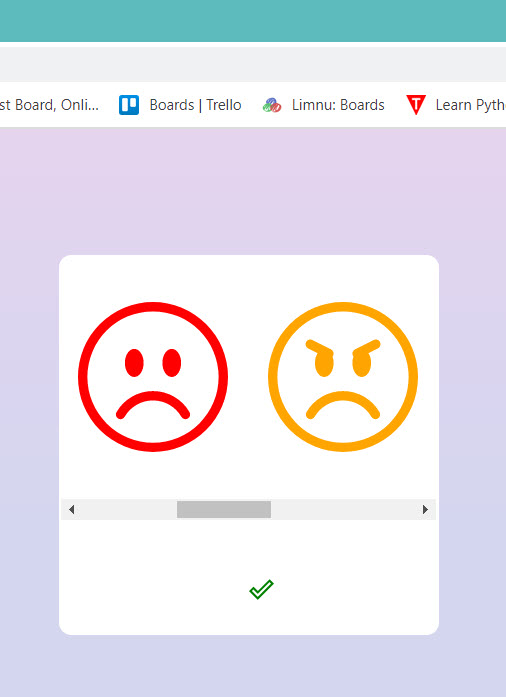
از آنجایی که از ویژگی های css در این پروژه استفاده کرده‌ایم، ریسپانسیو بودن یکی از مزایای این پروژه است که با استفاده از Css آنرا به نمایش گذاشته ایم.

تصویر4-23 کامپوننت Water

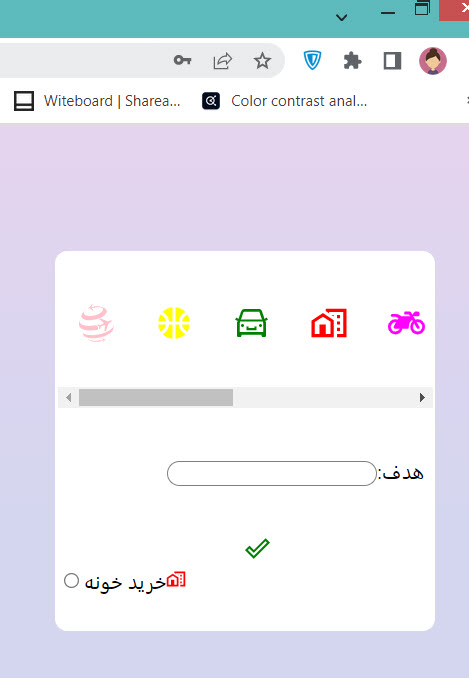


تصویر4-24 بخشی از کامپوننت Emotions

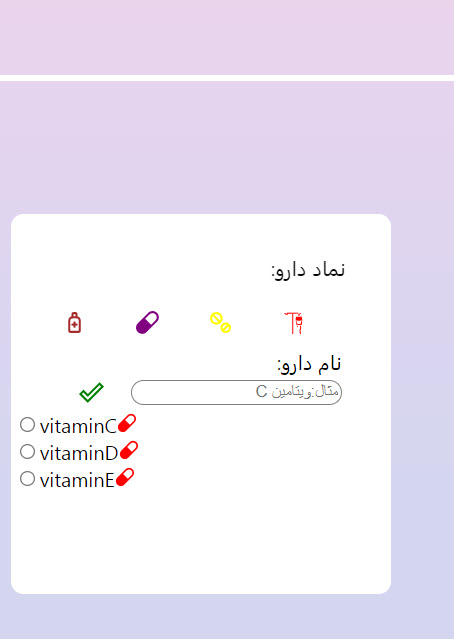
تصویر4-25 بخشی از کامپوننت Emotions



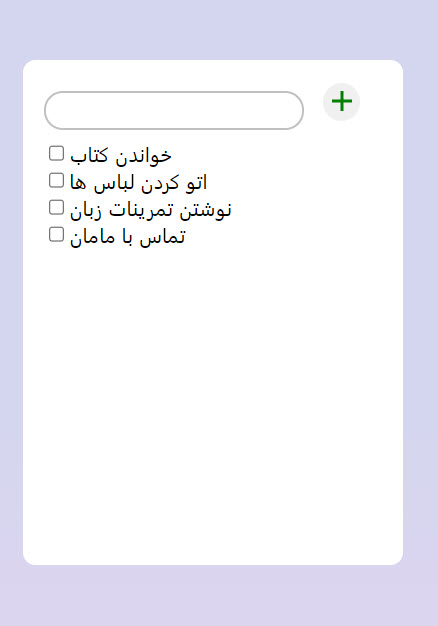
تصویر4-26 بخشی از کامپوننت Emotions

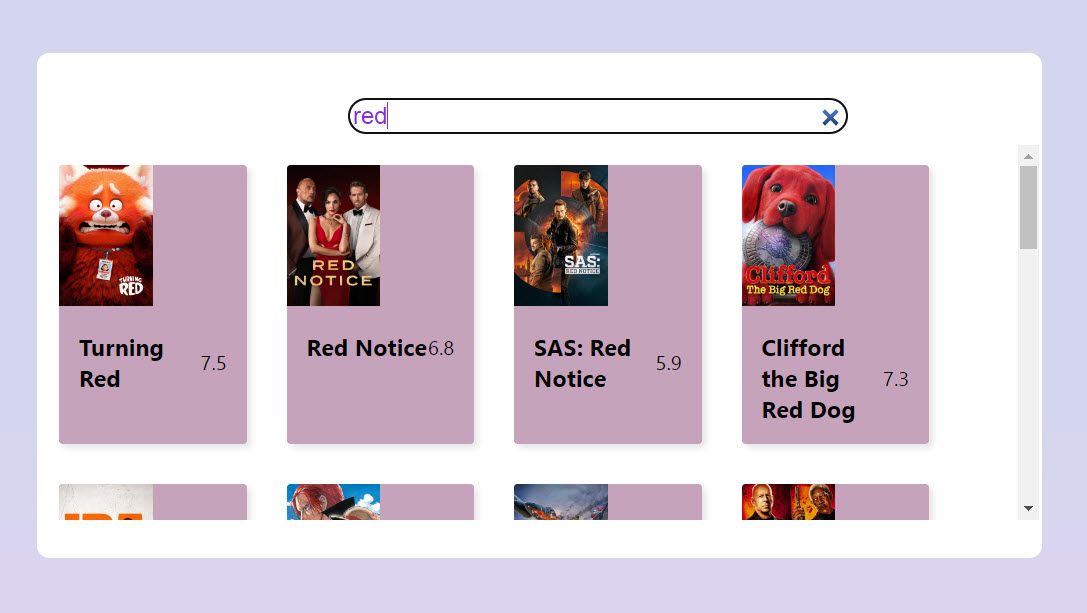


تصویر4-27 بخشی از کامپوننت Purposes

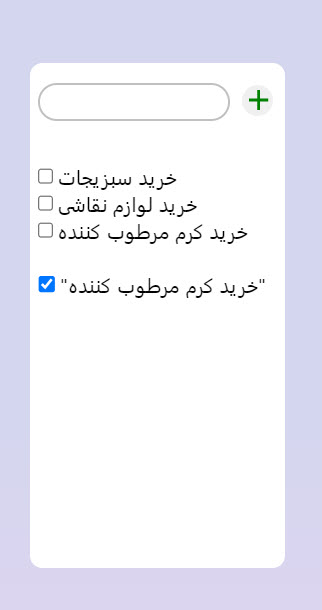


تصویر4-28 کامپوننت Medicions

تصویر4-29 کامپوننت TodoList



تصویر4-30 کامپوننت Movie



تصویر4-31 کامپوننت Shoplist



تصویر 4-32 کامپوننت Calender به همراه Api تقویم فارسی

تصویر4-33 کامپوننت ThanksFor



تصویر4-34 کامپوننت Event

* 1. نتیجه‌گیری

در پایان این فصل قصد داریم درباره خلاصه ای از کارهایی که انجام داده ایم صحبت کنیم.

در این پژوهش سعی کرده ایم تا یک سایتی را با استفاده از روش های نوین که در حال حاضر در دنیای وب استفاده متفاوتی می‌شود را بیان کنیم. بنابراین سعی کردیم از روشی استفاده کنیم تا هم چشمگیر باشد و هم کارامد.

از این رو از زبان برنامه نویسی تحت سایت جاوااسکریپت و کتابخانه React که در این پژوهش هدف اصلی ما بود استفاده کردیم و با آن وب اپلیکیشنی را پیاده سازی کردیم که هم از طریق سایت و هم از طریق موبایل قابل استفاده باشد.

این کدنویسی ها را با استفاده از دستورات HTML و به صورت JSX در قالب Single Page Application به اجرا در آوردیم.

همچنین از کدهای Css و مفاهیم طراحی UI/UX در طراحی و زیبا سازی این پروژه بهره برده ایم.

بعلاوه سعی کردیم سایتی پویا ارائه کنیم بنابراین از انیمیشن ها در بخش های اصلی سایت که در واقع کامپوننت ها هستند استفاده نموده ایم و همچنین api هایی که کارها را برای کاربر ساده تر می‌کنند.

تمام این ویيگی ها را به کار برده ایم تا یک پروژه فرونت ارائه کنیم.

فرونت اند تمام کارخای یک سایت نیست بلکه ظاهری ساده و پر مفهوم برای کاربر ایجاد می‌کند.

در ادامه فصل بعد درباره پیشرفت پروژه صحبت خواهیم کرد.

اینکه چطور یک پلنر ارائه شده محبوب تر شود و چه ویژگی هایی به آن اضافه شود.

1. جمع‌بندی و پیشنهادها
   1. نتیجه­گیری

در این پژوهش سعی کردیم برنامه‌ای ارائه دهیم تا هر فرد بتواند کارهای روزمره خود را تحت آیتم هایی به ثبت برساند و از فراموش کردن آنها خود داری کند.

این کارها را با استفاده از زبان برنامه نویسی جاوااسکریپت انجام داده ایم و در قالب یک پروژه FrontEnd یک پلنری طراحی کرده ایم تا شخص بتواند کارهایی از جمله لیست کردن کارهای روز یا بیان احساسات خویش انجام دهد.

پروژه ای که ارائه کرده ایم تنها در قالب ظاهر یک سایت است و در جهت توسع آن نیازمند برنامه نویسی Backend داریم تا کارهایی فراتر از آن بتوان انجام داد.

اما سعی کردیم در این پروژه از منابعی استفاده کنیم تا ذخیره اطلاعت را برعهده گیرد و حتی یکسری خدمات به ما ارائه دهد.

در ادامه به روش هایی خواهیم پرداخت تا بتوان این پروژه را توسعه داد.

* 1. پیشنهادهایی برای کارهای آتی

در آینده قصد داریم خدماتی را ارائه کنیم که شخص بتواند برنامه های دیگری را به صفحه خویش اضافه کند.

همچنین به کاربران اجازه تغییر ظاهر سایت را بدهیم،آیکون ها زا تغییر دهد،آنرا تاریک و روشن کند،اندازه ها را تغییر دهد وآنهارا به شماره تماس یا ایمیل خود ارسال کند.

البته برای انجام این عملیات به تنهایی ممکن نیست بلکه نیازمند زبان های بک اند نیز می‌باشیم.

منابع

فهرست منابع

نمونه صفحات وب

1. <https://mui.com/material-ui/api/modal/>.
2. <https://stackoverflow.com/questions/>
3. <https://reactjs.org/docs/getting-started.html>
4. <https://github.com/ahmadpanah/ahmadpanah>
5. https://www.youtube.com/watch?v=gwj6HEkpjm0&list=PL6wpw9cKs6t9QJ2GC1jrKV81SVDVWZp1C

پیوست‌ها

پیوست الف (در صورت وجود)

Abstract

Research Aim: Primary purposes of research, and the research motivation or the reasons why you decided to do this study.

Research method: Describe approaches, methods or techniques whose explains are crucial to understanding the study.

Findings: Findings should be written in a concise and comprehensive manner. This includes experimental and theoretical findings, relations and correlations, observed impacts, etc.

Conclusion: Based on the results of your study, this section should explain the contributions and implications of the research. It may also include suggestions for further studies, evaluations, applications, new relations, and approved & rejected hypothesis.

**Keywords:** Insert your research keywords here.



Technical and Vocational University

Shariaty Technical College

A Dissertation Submitted in Partial Fulfillment for the Degree of Bachelor of Science in Elecrical/ICT/Software Engineering

Thesis Title

By:

Name of Student

Supervisor:

Name of Supervisor

Month Year

1. Web Application [↑](#footnote-ref-1)
2. Application [↑](#footnote-ref-2)
3. FrontEnd [↑](#footnote-ref-3)
4. BackEnd [↑](#footnote-ref-4)
5. C# [↑](#footnote-ref-5)
6. PHP [↑](#footnote-ref-6)
7. Python [↑](#footnote-ref-7)
8. JavaScript [↑](#footnote-ref-8)
9. Libraries [↑](#footnote-ref-9)
10. Hyper Text Markup Language [↑](#footnote-ref-10)
11. Cascade StyleSheet [↑](#footnote-ref-11)
12. G-mail [↑](#footnote-ref-12)
13. User Interface [↑](#footnote-ref-13)
14. User Exprience [↑](#footnote-ref-14)
15. Single Page Application [↑](#footnote-ref-15)
16. input [↑](#footnote-ref-16)
17. Modal [↑](#footnote-ref-17)
18. router [↑](#footnote-ref-18)
19. link [↑](#footnote-ref-19)
20. **J**ava**S**cript **O**bject **N**otation [↑](#footnote-ref-20)
21. Application Programming Interface [↑](#footnote-ref-21)
22. EcmaScript5 [↑](#footnote-ref-22)
23. Facebook [↑](#footnote-ref-23)
24. G-mail [↑](#footnote-ref-24)
25. https://nodejs.org/ [↑](#footnote-ref-25)
26. code.visualstudio.com [↑](#footnote-ref-26)
27. Framework [↑](#footnote-ref-27)
28. git-scm.com [↑](#footnote-ref-28)
29. Component [↑](#footnote-ref-29)
30. id [↑](#footnote-ref-30)
31. JavaScript Xml [↑](#footnote-ref-31)
32. Modal [↑](#footnote-ref-32)
33. َApplication Programing Interface [↑](#footnote-ref-33)