



دانشگاه فنی و حرفه‌ای  
دانشکده فنی و حرفه‌ای پسران شهید شمس‌پور  
رشته کامپیوتر، گرایش مهندسی فناوری اطلاعات

پایان‌نامه برای دریافت درجه کارشناسی

عنوان پروژه

# **کاربرد لاراول در حوزه سلامت الکترونیک، مطالعه موردی کالری شمار**

استاد راهنما

علی متین وفا

نگارنده

ارسلان ارغون

تاریخ ارائه:

تابستان ۱۴۰۲

بِه نامِ خداوندِ  
جان و خیرد

## تقدیر و تشکر

بسیار خوشحالم که با کمک استاد راهنمای عزیزم، پایان‌نامه‌ام به بهترین شکل ممکن به پایان رسید. استاد راهنمای عزیزم آقای علی متین وفا با دانش و تجربه‌ی بالای خود، به من کمک کرد تا به بهترین شکل ممکن پژوهش خود را انجام دهم و نتیجه خوبی را به دست آورم.

از استاد راهنمای خود به خاطر تلاش‌های بی‌دریغ و راهنمایی‌های ارزشمندی که برایم انجام دادند، صمیمانه تشکر می‌کنم. یکی از بهترین استادانی است که با انگیزه و عشق به حرفه خود، دانشجویان خود را به بهترین شکل آموزش می‌دهند و به آن‌ها انگیزه می‌دهند تا برای دستیابی به اهدافشان تلاش کنند.

باز هم از استاد عزیزم به خاطر تمامی تلاش‌هایی که برایم انجام دادند و راهنمایی‌های ارزشمندی که به من دادند، تشکر می‌کنم.

## فهرست

۱	تقدیر و تشکر.....
۱	چکیده .....
۳	۱.۱. مقدمه .....
۴	۱.۲. بیان مسئله .....
۴	۱.۳. فرضیات .....
۵	۱.۴. اهداف پژوهش .....
۵	۱.۵. روش پژوهش .....
۸	۲.۱. معرفی سایت .....
۸	۲.۲. بخش‌های کاربری سیستم .....
۹	۲.۳. زبان مدل‌سازی یکنواخت (UML) .....
۹	۲.۳.۱. تاریخچه UML .....
۱۰	۲.۳.۲. خصوصیات UML .....
۱۰	۲.۳.۳. مراحل مدل‌سازی به روش UML .....
۱۱	۲.۳.۴. نمودارها در UML .....
۱۲	۲.۳.۴.۱. تعریف نمودار مورد کاربرد Use Case Diagram .....
۱۲	۲.۳.۴.۲. تعریف نمودار Class Diagram .....
۱۳	۲.۳.۴.۳. تعریف نمودار Sequence .....
۱۳	۲.۳.۴.۴. تعریف نمودار Collaboration .....

۱۳	۲.۳.۴.۵. تعریف نمودار فعالیت Activity Diagram
۱۳	۲.۳.۴.۶. تعریف نمودار Component
۱۴	۲.۳.۴.۷. تعریف نمودار Deployment
۱۴	۲.۴. مدل کسب و کار
۱۵	۲.۴.۱. مدل Use Case کسب و کار
۱۵	۲.۴.۱.۱. نمودار Use Case مدیر سایت
۱۵	۲.۴.۱.۲. نمودار Use Case کاربر
۱۶	۲.۴.۲. نمودار فعالیت
۱۷	۲.۴.۲.۱. ثبت نام
۱۷	۲.۴.۲.۲. ورود
۱۸	۲.۴.۲.۳. افزودن محصول جدید
۱۸	۲.۴.۲.۴. افزودن ظرف جدید
۱۹	۲.۴.۲.۵. افزودن محصول به ظرف
۱۹	۲.۴.۲.۶. افزودن منوی جدید
۲۱	۳.۱. احراز هویت، ورود به سیستم و سطح دسترسی‌ها
۲۱	۳.۲. تنظیمات اولیه
۲۱	۳.۲.۱. نصب لاراول
۲۱	۳.۲.۲. نصب سرور مجازی
۲۱	۳.۲.۳. ساخت دیتابیس

۲۲	۳.۲.۴. اجرای پروژه.....
۲۳	۳.۳. دیتابیس این پروژه.....
۲۳	۳.۳.۱. مایگریشن.....
۲۴	۳.۴. میان افزار ( MiddleWare) .....
۲۵	۳.۵. قالب وبسایت.....
۲۶	۳.۶. لاراول میکس.....
۲۶	۳.۷. ایجاد محصولات.....
۲۹	۳.۸. مسیرها (Routeها).....
۳۲	۳.۹. بخش احراز هویت.....
۳۶	۴.۱. صفحات اصلی سایت.....

## چکیده

پروژه کالری شمار آنلاین با استفاده از Laravel و Vue.js یک برنامه وب است که به کاربران امکان محاسبه کالری مصرفی، ثبت میزان مصرف غذا و ورزش، و مشاهده آمار و گزارش‌های مربوط به سلامتی را می‌دهد. این برنامه با استفاده از Laravel که یک چارچوب PHP قدرتمند برای توسعه برنامه‌های وب است و Vue.js که یک فریم‌ورک جاوا اسکریپت برای ساخت رابط کاربری است، طراحی شده است.

ویژگی‌های این برنامه عبارتند از:

- محاسبه کالری مصرفی در یک روز بر اساس وزن، قد، سن، جنسیت و سطح فعالیت کاربر
- ثبت میزان مصرف غذا و ورزش روزانه توسط کاربر
- نمایش آمار و گزارش‌های مربوط به میزان مصرف کالری، میزان مصرف غذا و ورزش و وزن کاربر در یک بازه زمانی مشخص
- امکان تنظیم رژیم غذایی و ورزشی مناسب برای کاربران
- ارائه محتوای آموزشی مرتبط با سلامتی و تغذیه به کاربران
- پشتیبانی از زبان‌های مختلف

در کل، پروژه کالری شمار آنلاین با استفاده از Laravel و Vue.js یک برنامه جامع و قدرتمند برای محاسبه کالری مصرفی و مراقبت از سلامتی کاربران است که با استفاده از تکنولوژی‌های پیشرفته، امکانات مختلفی را به کاربران ارائه می‌دهد.

# فصل اول

کلیات تحقیق



## ۱.۱. مقدمه

در دنیای امروز، با پیشرفت تکنولوژی و انتشار فضای دیجیتال، حوزه سلامت الکترونیک به عنوان یکی از موضوعات مهم و پرطرفدار در بسیاری از کشورها به شمار می‌رود. این حوزه با استفاده از فناوری‌های نوین، امکانات جدیدی را برای بیماران، پزشکان و سایر متخصصان حوزه سلامت فراهم کرده است. یکی از این امکانات، سامانه کالری شمار است که برای تعیین میزان کالری مصرفی در غذاها بسیار مفید است. در این پایان نامه، به بررسی کاربرد فریم‌ورک لاراول در توسعه سامانه کالری شمار پرداخته‌ایم.

سامانه کالری شمار، با توجه به اهمیت مصرف سالم و مناسب غذا، به یکی از مهمترین سامانه‌های حوزه سلامت الکترونیک تبدیل شده است. استفاده از سامانه کالری شمار، به کاربران این امکان را می‌دهد تا با دقت بیشتری به میزان کالری‌های مصرفی خود توسط غذاهای مختلف دقت کنند. با توجه به اینکه استفاده از فناوری‌های نوین در حوزه سلامت الکترونیک به یکی از موضوعات پرطرفدار در دنیای امروز تبدیل شده است، در این پایان نامه به بررسی کاربرد فریم‌ورک لاراول در توسعه سامانه کالری شمار پرداخته شده است.

یکی از مزایای استفاده از فریم‌ورک لاراول، سرعت بالا و قابلیت پذیرش بالا در برنامه‌های وب است. در این پایان نامه، علاوه بر بررسی مزایا و معایب استفاده از لاراول، به بررسی و مقایسه با فریم‌ورک‌های دیگر نیز پرداخته خواهد شد. هدف این پایان نامه، آشنایی با فریم‌ورک لاراول و بررسی میزان کارآمدی آن در توسعه سامانه کالری شمار است.

این پایان نامه با هدف بررسی چگونگی استفاده از لاراول برای توسعه سامانه کالری شمار، به عنوان یکی از مهمترین ابزارهای حوزه سلامت الکترونیک، انجام شده است. در این پایان نامه، ابتدا به بیان مفهوم حوزه سلامت الکترونیک و اهمیت آن در دنیای امروز پرداخته شده است. همچنین، به بررسی مفهوم سامانه کالری شمار و اهمیت آن برای تعیین میزان کالری مصرفی در غذاها پرداخته شده است.

در پایان پایان نامه، نتایج به دست آمده از بررسی و مقایسه لاراول با فریم‌ورک‌های دیگر در توسعه سامانه کالری شمار، ارائه شده است.

## ۱.۲. بیان مسئله

با توجه به نیاز به توسعه پلتفرمی قابل اعتماد، کارآمد و همچنین ساده برای محاسبه دقیق کالری مصرفی افراد، به خصوص افرادی که مشکلاتی مانند چاقی یا کمبود وزن دارند، مسئله طراحی و پیاده‌سازی یک پروژه لاراول در حوزه سلامت الکترونیک، کالری شمار به عنوان یکی از مهم‌ترین وظایف در این حوزه، جدی‌تر از گذشته به نظر می‌رسد. در حال حاضر، چالش‌هایی در زمینه کالری شمار وجود دارد که نیاز به توسعه راهکارهای نوین جهت حل این چالش‌ها را احساس می‌کنند. به همین دلیل، در این پایان‌نامه، با بررسی موضوع کالری شمار و چالش‌های آن، به طراحی و پیاده‌سازی یک پروژه لاراول در حوزه سلامت الکترونیک، کالری شمار، پرداخته‌ایم.

سوالات تحقیقی این پایان‌نامه، شامل موارد زیر هستند:

- آیا پروژه لاراول در حوزه سلامت الکترونیک، کالری شمار به صورت یک پلتفرم قابل اطمینان، کارآمد و ساده، قابل پیاده‌سازی است؟
- آیا پروژه لاراول در حوزه سلامت الکترونیک، کالری شمار، قابلیت ارائه دقیق‌ترین محاسبات کالری مصرفی را دارد؟
- آیا پروژه لاراول در حوزه سلامت الکترونیک، کالری شمار، قابلیت سازگاری با دستگاه‌های مختلف و انواع غذاها و نوشیدنی‌ها را دارد؟
- آیا کاربران برای استفاده از پروژه لاراول در حوزه سلامت الکترونیک، کالری شمار، به راحتی می‌توانند با آن ارتباط برقرار کنند و از آن استفاده کنند؟

با پاسخ به این سوالات، در این پایان‌نامه، به بررسی و رفع چالش‌های موجود در زمینه کالری شمار و همچنین به طراحی و پیاده‌سازی یک پروژه لاراول در حوزه سلامت الکترونیک، کالری شمار، پرداخته شده است.

## ۱.۳. فرضیات

- فرضیه ۱: داده‌های جمع‌آوری شده در این تحقیق، به‌طور کامل، دقیق و صحیح هستند.
- فرضیه ۲: پروژه لاراول در حوزه سلامت الکترونیک، کالری شمار، قابلیت ارائه دقیق‌ترین محاسبات کالری مصرفی را دارد.

- فرضیه ۳: پروژه لاراول در حوزه سلامت الکترونیک، کالری شمار، قابلیت سازگاری با دستگاه‌های مختلف و انواع غذاها و نوشیدنی‌ها را دارد.
- فرضیه ۴: کاربران برای استفاده از پروژه لاراول در حوزه سلامت الکترونیک، کالری شمار، به راحتی می‌توانند با آن ارتباط برقرار کنند و از آن استفاده کنند.

## ۱.۴. اهداف پژوهش

- بررسی دقت و کارایی الگوریتم‌های مختلف کالری شماری در برنامه‌های سلامت الکترونیکی موجود: در این پژوهش، از الگوریتم‌های مختلف کالری شماری استفاده خواهد شد و دقت و کارایی هر الگوریتم را در یک برنامه سلامت الکترونیکی بررسی می‌کنیم. می‌توانیم به دنبال بهینه‌سازی الگوریتم‌های موجود برای بهبود دقت و کارایی آن‌ها باشیم.
- توسعه یک برنامه کالری شمار سلامت الکترونیکی: این پژوهش در واقع به توسعه یک برنامه کالری شمار سلامت الکترونیکی می‌پردازد که می‌تواند به کاربران کمک کند تا کالری‌های مصرفی خود را به راحتی ردیابی کنند. در این پژوهش، می‌توان به بهبود رابط کاربری، دقت کالری شماری و ارائه ویژگی‌های جدید برای برنامه پرداخت.
- بررسی تأثیر برنامه کالری شماری سلامت الکترونیکی بر سلامت فرد: در این پژوهش، می‌توان به بررسی تأثیر استفاده از یک برنامه کالری شمار سلامت الکترونیکی بر سلامت فرد پرداخت. برای مثال، می‌توان به بررسی تأثیر کالری شماری در کاهش وزن یا بهبود سلامت قلبی عروقی پرداخت.
- بررسی عوامل مؤثر در موفقیت برنامه کالری شماری سلامت الکترونیکی: در این پژوهش، می‌توان به بررسی عواملی مانند تعامل کاربر با برنامه، میزان تعهد فرد به استفاده از برنامه، ویژگی‌های رابط کاربری و ... پرداخته و تأثیر آن‌ها بر موفقیت برنامه کالری شماری را بررسی نمود.

## ۱.۵. روش پژوهش

در این پایان‌نامه، با استفاده از روش تحقیق کمی، به بررسی عملکرد سایت نیتروی سلامتی به عنوان یک کالری شمار پرداخته شده است. هدف این پایان‌نامه بررسی دقت و کارایی سایت نیتروی سلامتی در شرایط مختلف و با توجه به داده‌های جمع‌آوری شده از کاربران و سیستم‌های مختلف است.

برای انجام این پایان‌نامه، از روش تحقیق کمی استفاده شده است. در این روش، با جمع‌آوری داده‌های عددی و آماری از کاربران و سیستم‌های مختلف، به بررسی کارایی و دقت سایت پرداخته شده است. برای جمع‌آوری داده‌ها، از پرسشنامه‌های کمی و نظرسنجی‌های آنلاین استفاده شده است.

نتایج بررسی نشان داد که سایت نیتروی سلامتی به طور کلی، دارای کارایی و دقت قابل قبولی است. با این حال، برخی مشکلات و نواقص در عملکرد سایت در شرایط خاص مشاهده شده است که باید با دقت بیشتری مورد بررسی قرار گیرد.

به طور کلی، با توجه به نتایج به دست آمده، می‌توان گفت که سایت نیتروی سلامتی به عنوان یک کالری شمار دارای کارایی و دقت قابل قبولی است.

# فصل دوم

توضیحات پروژه

## ۲.۱. معرفی سایت

نیتروی سلامتی یک پروژه وب است که به کاربران امکان محاسبه کالری مصرفی در یک روز، محاسبه BMI، دریافت رژیم غذایی و ورزشی مناسب، و فروشگاه اینترنتی برای خرید لوازم ورزشی را می‌دهد. این پروژه با هدف بهبود سلامتی افراد ایجاد شده و با استفاده از الگوریتم‌های پیچیده‌ای که بر اساس اطلاعاتی مانند وزن، قد، سن، جنسیت و سطح فعالیت، کالری مصرفی روزانه را محاسبه می‌کند.

با محاسبه کالری مصرفی، کاربران می‌توانند بهترین راهکار برای کاهش وزن یا حفظ وزن سالم خود را پیدا کنند. همچنین، با محاسبه BMI، کاربران می‌توانند وضعیت بدنی خود را بررسی کنند و با استفاده از رژیم غذایی و ورزشی مناسب، به بهترین شکل ممکن برای سلامتی خود مراقبت کنند.

همچنین، این پروژه به کاربران امکان دسترسی به رژیم غذایی و ورزشی مناسب را به همراه توضیحات آن‌ها می‌دهد. با استفاده از این رژیم‌ها، کاربران می‌توانند به بهترین شکل ممکن برای سلامتی خود مراقبت کنند و به راحتی رژیم‌های غذایی و ورزشی خود را تغییر دهند.

همچنین، نیتروی سلامتی به کاربران امکان دسترسی به فروشگاه اینترنتی برای خرید لوازم ورزشی را نیز می‌دهد. این فروشگاه به کاربران این امکان را می‌دهد تا به راحتی لوازم ورزشی مورد نیاز خود را تهیه کنند و به بهترین شکل ممکن برای سلامتی خود مراقبت کنند.

در کل، پروژه نیتروی سلامتی یک پروژه وب است که به کاربران امکانات بیشتری برای بهبود سلامتی خود می‌دهد و با استفاده از الگوریتم‌های پیچیده، به بهترین شکل ممکن برای محاسبه کالری مصرفی و ارائه رژیم غذایی و ورزشی مناسب به کاربران خود، به بهبود سلامتی آن‌ها کمک می‌کند.

## ۲.۲. بخش‌های کاربری سیستم

- مدیر سیستم

- کاربر

## ۲.۳. زبان مدل‌سازی یکنواخت (UML)

تعریف UML: زبان UML اختصاری برای عبارت «زبان مدل‌سازی یکپارچه» (Unified Modeling Language) است که یک زبان مدل‌سازی چندمنظوره محسوب می‌شود. هدف اصلی UML تعریف یک روش استاندارد برای بصری‌سازی مسیر طراحی سیستم‌ها است. UML شباهت زیادی به نقشه‌های اولیه‌ای دارد که در رشته‌های دیگر مهندسی مورد استفاده قرار می‌گیرند.

توجه داشته باشید که UML یک زبان برنامه‌نویسی نیست، بلکه یک زبان بصری‌سازی محسوب می‌شود. ما از نمودارهای UML برای نمایش رفتار و ساختار یک سیستم استفاده می‌کنیم. UML به مهندسان نرم‌افزار، تاجران و معماران سیستم در زمینه‌های مدل‌سازی، طراحی و تحلیل کمک می‌کند. از این رو آموزش UML به همه این افراد یک امر ضروری محسوب می‌شود. «گروه مدیریت شیء» (Object Management Group) به اختصار OMG در سال ۱۹۹۷، UML را به عنوان یک استاندارد پذیرفته است. از آن زمان تاکنون این زبان از سوی OMG مدیریت می‌شود. UML در سال ۲۰۰۵ از سوی ISO به عنوان استاندارد انتشار یافته است و در طی سال‌های اخیر با تحولات زیادی مواجه گشته است.

### ۲.۳.۱. تاریخچه UML

قبل از ورود به بحث آموزش UML به بررسی مختصری از تاریخچه آن می‌پردازیم. دهه ۱۹۹۰ عصر توسعه زبان‌های برنامه‌نویسی شیء‌گرا از قبیل C++ بود. این زبان‌های شیء‌گرا برای ایجاد سیستم‌های پیچیده و قدرتمند مورد استفاده قرار می‌گرفتند. به مرور که درک این سیستم‌ها دشوار شد، برخی مشکلات طراحی و تحلیل در فرایند توسعه سیستم ایجاد گشت. مشکل این بود که توضیح سیستم به افراد دیگر دشوار شده بود.

به محض ورود UML بسیاری از آزمایش‌ها و رویکردهای تحول‌بخش برای ساده‌سازی چنین وظایف دشوار تحلیل سیستم به اجرا درآمدند. UML یک زبان مدل‌سازی یکپارچه شیء‌گرا است که از سوی مهندسان نرم‌افزار به نام‌های «گریدی بوچ» (Grady Booch)، «ایوار جیکوبسون» (Ivar Jacobson) و «جیمز رامباو» (James Rumbaugh) از شرکت نرم‌افزاری Rational در طی سال‌های ۱۹۹۴ و ۱۹۹۵ طراحی شد و از آن زمان تاکنون در حال توسعه است. هر یک از این مهندسان ایده‌های جالبی برای طراحی زبانی داشتند که پیچیدگی سیستم‌ها را کاهش دهند.

## ۲.۳.۲. خصوصیات UML

- یک زبان مدل سازی عمومی است.
- از زبان های برنامه نویسی مانند پایتون، C یا C++ و غیره متفاوت است.
- یک زبان تصویری است که می تواند برای تولید عناصر مدل سازی قدرتمند مورد استفاده قرار گیرد.
- به طراحی و تحلیل شیءگرایی مرتبط است.
- کاربردهای نامحدودی حتی خارج از صنعت نرم افزار دارد و می تواند برای بصری سازی گردش کار یک کارخانه نیز مورد استفاده قرار گیرد.

## ۲.۳.۳. مراحل مدل سازی به روش UML

۱. متن ساده ای می نویسیم.
۲. زیر قسمت های مهم متن خط میکشیم.
۳. هر کدام از خطوط کشیده شده به یک Business Use Case تبدیل می شود.
۴. کلیه Business Use Case ها و Business Actor ها را مشخص می کنیم.
۵. نمودار Business Use Case Modeling را رسم می کنیم.
۶. به تعداد Business Actor ها یک نمودار Business Use Case Work Flow رسم می کنیم.
۷. از نمودار مرحله ۶، Business Worker و Business Entity را استخراج می کنیم.
۸. نمودار Business Object Model را با استفاده از نتیجه مرحله ۶ رسم می کنیم.
۹. با استفاده از دو نمودار قبلی Business Use Case Diagram و Business Object Modeling یک نمودار Use Case Diagram را می کشیم.
۱۰. از Use Case Diagram ها دو شکل Actor و Use Case حاصل می شود.



۱۱. می‌توان برای هر Use Case یک Diagram Activity رسم کرد.
۱۲. نمودار کلاس مفهومی Conceptual Class را از Object Modeling به دست آورده و سپس Class Diagram را رسم می‌کنیم.
۱۳. Boundary Class را از Use Case به دست می‌آوریم.
۱۴. چگونگی طراحی فرم Story Board را از Boundary Class و Use Case به دست می‌آوریم.
۱۵. نمودار سه لایه‌ای Modeling مان را رسم می‌کنیم.
۱۶. Modeling Sequence را رسم می‌کنیم.
۱۷. در قسمت مدل‌سازی باید برای تک‌تک موارد کاربرد کسب و کار Business Use Case که در قسمت قبل شناسایی شد جریان کار را به وسیله یک نمودار فعالیت شرح دهیم.
۱۸. Business Worker: کسی است که با این سیستم کار می‌کند.
۱۹. Business Actor: کسی است که سیستم برای آن ایجاد شده است.
۲۰. Business Entity: خروجی مستندات و اشیایی که حاوی اطلاعات هستند.

## ۲.۳.۴. نمودارها در UML

- نمودار مورد کاربرد (Use Case Diagram)
- نمودار کلاس (Class Diagram)
- نمودار تعامل (Interaction Diagram)
- نمودار توالی (Sequence Diagram)
- نمودار همکاری (Collaboration Diagram)
- نمودار رفتار (Behavior Diagram)
- نمودار فعالیت (Activity Diagram)
- نمودار حالت (State Diagram)
- نمودار پیاده‌سازی (Implementation Diagram)
- نمودار اجزا (Component Diagram)

- نمودار استقرار (Deployment Diagram)

## ۲.۳.۴.۱. تعریف نمودار مورد کاربرد Use Case Diagram

این نمودار با هدف نمایش وظایف کلی سیستم و کاربران آن ایجاد می‌شود به دلیل اینکه این نمودار ساختار ساده‌ای دارد و در آن از علائم کم و ساده‌ای استفاده می‌شود. نمودار مورد کاربرد برای ایجاد یک درک متقابل بین تحلیل‌گر و کاربر سیستم مورد استفاده قرار می‌گیرد. یک Use Case مجموعه‌ای از فعالیت‌ها که منجر به حصول یک نتیجه خاص برای سیستم می‌شود. در سیستم دانشگاه مثل ثبت‌نام، حذف و اضافه، انتخاب واحد، حذف اضطراری. دو نوع Actor داریم؛ خدمات‌دهنده و خدمات‌گیرنده که از سیستم استفاده می‌کنند.

اهداف و مقاصد از تهیه نمودار Use Case Diagram عبارتند از:

- ایجاد یک دید سطح بالا از آنچه سیستم انجام می‌دهد و کسانی که سیستم را به کار می‌گیرند.
- شناسایی مجموعه یا توالی فعالیت‌هایی که اجرای آنها منجر به ایجاد یک مزیت یا منفعت برای کاربران سیستم می‌شود.
- این مدل می‌تواند به عنوان مبنایی برای طراحی تعامل میان کاربر و نرم افزار قرار گیرد.
- این مدل می‌تواند برای شناسایی شیوه‌های جایگزینی مختلفی که برای انجام فعالیت سیستم وجود دارد مورد استفاده قرار گیرد.

## ۲.۳.۴.۲. تعریف نمودار Class Diagram

نمودار کلاس، نموداری است که کلاس‌ها و روابط Static میان آن‌ها را نشان می‌دهد. در نمودار کلاس، هدف نمایش کلاس‌ها و روابط آنها بدون توجه به عملیات سیستم می‌باشد.

اهداف و مقاصد از تهیه نمودار Class Diagram به شرح زیر هستند:

- نمودار کلاس می‌تواند در مراحل مختلف سیستم مورد استفاده قرار بگیرد.
- نمودار کلاس می‌تواند به منظور نمایش کلاس و روابط آن با کلاس‌های دیگر سیستم مورد استفاده قرار گیرد.

- در نمودار کلاس می‌توان یک شیء (Object) و ارتباط آن با مجموعه کلاس‌های آن سیستم را نمایش داد.
- مستند سازی کلاس‌های تشکیل دهنده سیستم یا زیر سیستم.

### ۲.۳.۴.۳. تعریف نمودار Sequence

نمودار توالی (Sequence) به صورت منظم و در یک توالی زمانی پشت سر هم ارتباطات متقابل اشیاء را به ما نشان می‌دهد. این دیاگرام برای انجام عمل خاصی در یک Use Case مشخص، مراحل انجام کار را مرحله به مرحله به شما نشان می‌دهد، یعنی دنباله‌ای از رویدادها را برای انجام یک عمل مشخص می‌سازد. در مراحل تحلیل و طراحی برای فهم نحوه عملکرد سیستم از این دیاگرام استفاده می‌شود. این نمودار در ارتباط مستقیم با نمودار همکاری می‌باشد.

### ۲.۳.۴.۴. تعریف نمودار Collaboration

نمودار همکاری، در طول توسعه نرم‌افزار به روش شیء‌گرا هر چیزی که نرم‌افزار نهایی نیاز داشته باشد به وسیله همکاری اشیاء صورت خواهد گرفت. می‌توان نمودار همکاری را برای تشریح چگونگی وضعیت اشیاء در حال همکاری، به کار برد. نمودار همکاری دقیقاً همان اطلاعات نمودارهای توالی را نشان می‌دهند با این تفاوت که Objectها و فعل و انفعالات عامل‌ها را بدون توجه به زمان نشان می‌دهد.

### ۲.۳.۴.۵. تعریف نمودار فعالیت Activity Diagram

نمودار فعالیت ابزاری است برای تشریح جریان کار (Work Flow) که می‌تواند مراحل چرخه حیات سیستم از تحلیل کسب و کار گرفته تا مراحل طراحی و پیاده‌سازی سیستم را نشان دهد. Activity Diagram بر اساس کاری است که انجام می‌دهیم.

### ۲.۳.۴.۶. تعریف نمودار Component

نمودارهای اجرا، یک دید فیزیکی از مدلتان را به شما نشان می‌دهد. یک نمودار Component اجزای نرم‌افزاری سیستم شما و روابط بین آن‌ها را به شما نمایش می‌دهد. این دیاگرام، نمایی فیزیکی از سیستم را در اختیار می‌گذارد. هدف این دیاگرام نشان دادن Dependencyهایی است که یک نرم‌افزار روی Componentهای نرم‌افزار دیگر در

سیستم دارد. می‌توان این دیاگرام را در سطحی بسیار بالا با Component های دانه درشت نشان داد یا می‌توان در سطح بسته (Component Package Level) نشان داد.

## ۲.۳.۴.۷. تعریف نمودار Deployment

این دیاگرام نشان می‌دهد یک سیستم چگونه به طور فیزیکی در محیط سخت‌افزار نصب می‌شود. هدفش نشان دادن جایی است که Component های مختلف سیستم به طور فیزیکی اجرا می‌شوند و چگونه با یکدیگر در ارتباط هستند. از آنجایی که این دیاگرام Runtime فیزیکی را مدل‌سازی می‌کند، پرسنل یک سیستم، استفاده قابل ملاحظه‌ای از این دیاگرام می‌کنند. علامت‌گذاری در Deployment Diagram شامل همان عناصر نشانه‌گذاری‌های Component Diagram، به علاوه چند علامت دیگر مثل مفهوم یک Node است؛ Node یا یک ماشین فیزیکی یا یک ماشین مجازی را نشان می‌دهد. برای مدل‌سازی یک Node، کافیسیت یک مکعب سه‌بعدی با نام Node در بالای مکعب، رسم شود.

## ۲.۴. مدل کسب و کار

طی دهه‌های اخیر، اتفاق مهمی افتاد و آن ظهور اینترنت بود. این ابزار جدید، فرصت‌های جدیدی را هم به وجود آورد و شیوه‌های درآمدزایی جدیدی هم خلق کرد.

قبلاً چند مدل کسب و کار مشخص وجود داشت و همه هم آن‌ها را می‌فهمیدند. اما حالا:

- شرکتی مثل یاهو (و بعداً گوگل) آمده بود که خدمات رایگان ارائه می‌داد.
  - آمازون کتاب‌ها را به شکلی می‌فروخت که در موارد زیادی، مجموع هزینه کتاب و حمل نقل، یا همان قیمت پشت جلد می‌شد و یا حتی کمتر از پشت جلد تمام می‌شد.
  - شرکت‌هایی مثل PayPal ایجاد شدند که به عنوان واسطه پولی عمل می‌کردند. اما به اندازه‌ی بانک‌ها سخت‌گیر نبودند و شرایطی فراهم شد تا کسب و کارهای آنلاین زیادی رشد کنند.
- امروزه وقتی کسی از مدل کسب و کار شما سوال می‌کند (یا وقتی خودتان می‌خواهید مدل کسب و کارتان را تعریف کنید)، تقریباً مشخص است که باید در پی پاسخ چه سوال‌هایی باشد.

می‌توان گفت مدل کسب و کار (Business Model) در تلاش برای یافتن پاسخ سه دسته سوال است:

- سوال‌های مربوط به ارزش آفرینی

○ سوال‌های مربوط به درآمدزایی

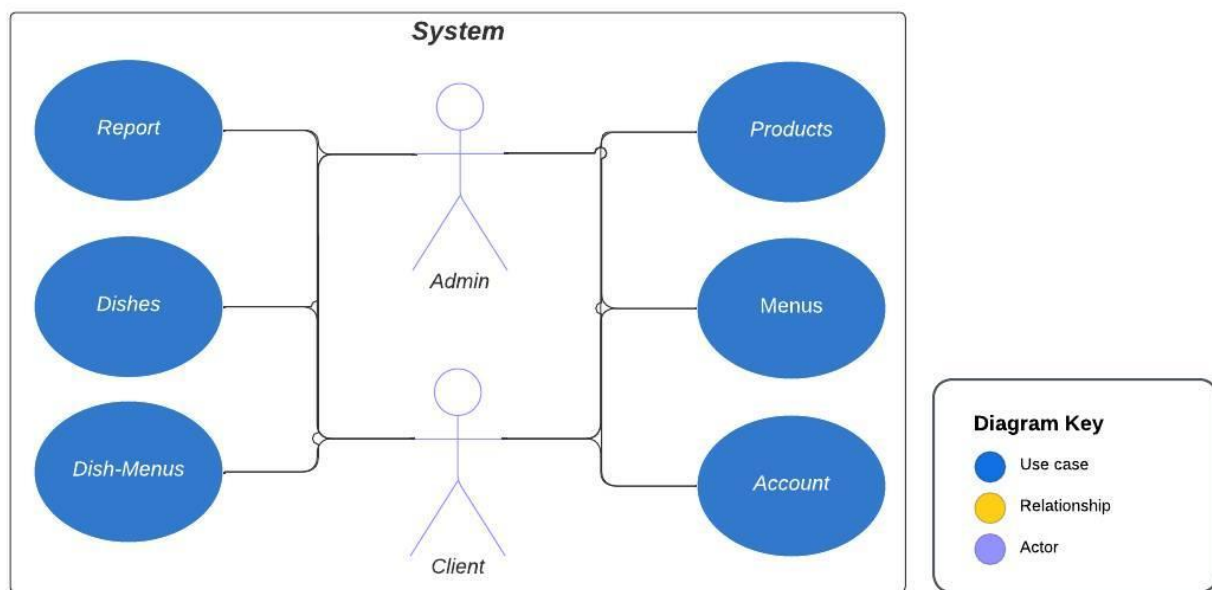
○ سوال‌های مربوط به پایداری و بقا

## ۲.۴.۱. مدل Use Case کسب و کار

در وب‌سایت نیتروی سلامتی که یک کالری شمار آنلاین است، دو دسترسی مدیر سایت و کاربر وجود دارد که هر کدام به بخش‌های مختلفی از وب‌سایت دسترسی دارند. مدیر وب‌سایت می‌تواند محصولات، رژیم‌های غذایی، ظرف‌ها و... را به سایت اضافه کرده و کاربر می‌تواند از محصولات آماده جهت کالری شماری استفاده نماید.

### ۲.۴.۱.۱. نمودار Use Case مدیر سایت

تمام دسترسی‌های مدیر سایت اعم از

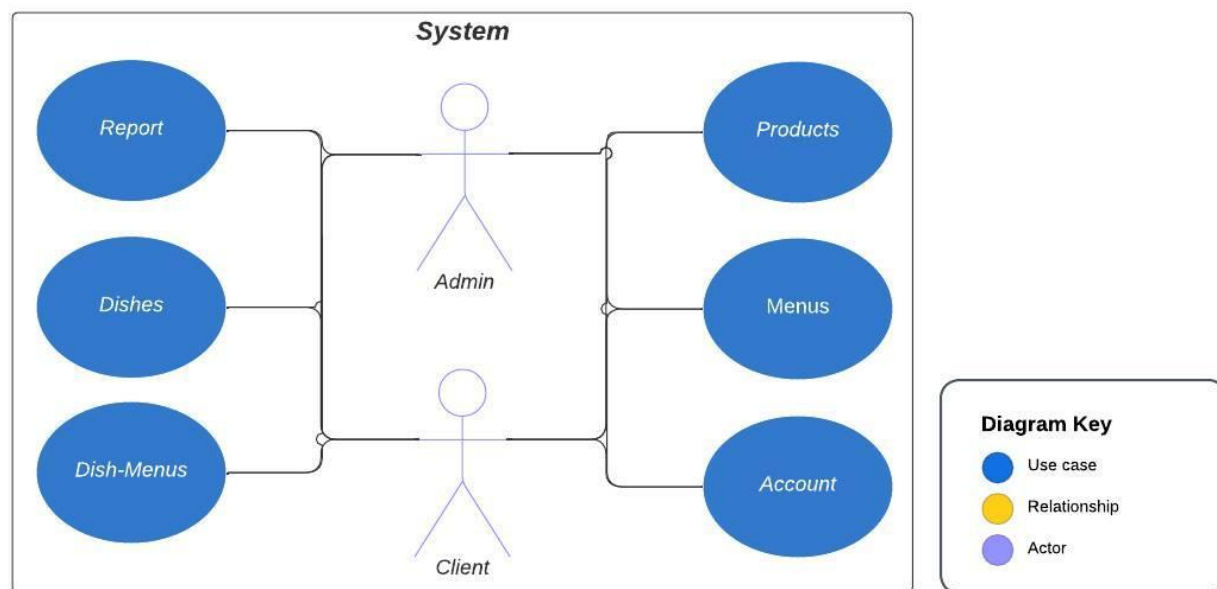


نمودار ۱- Use Case Diagram مدیر

### ۲.۴.۱.۲. نمودار Use Case کاربر

مشتری نیز دسترسی‌ها و امکانات مختلفی در اختیار دارد؛ مثلاً می‌تواند با مراجعه به سایت و مقایسه محصولات مختلف، محصول مورد نیاز خود را انتخاب کند و آن را به سبد خرید اضافه کند و با ثبت نام و ورود به سایت آدرس خود

را به پنل کاربری خود اضافه کند و یا محصولی را به لیست علاقه‌مندی‌های خود اضافه نماید و در نهایت اقدام به ثبت سفارش و پرداخت اینترنتی نماید.



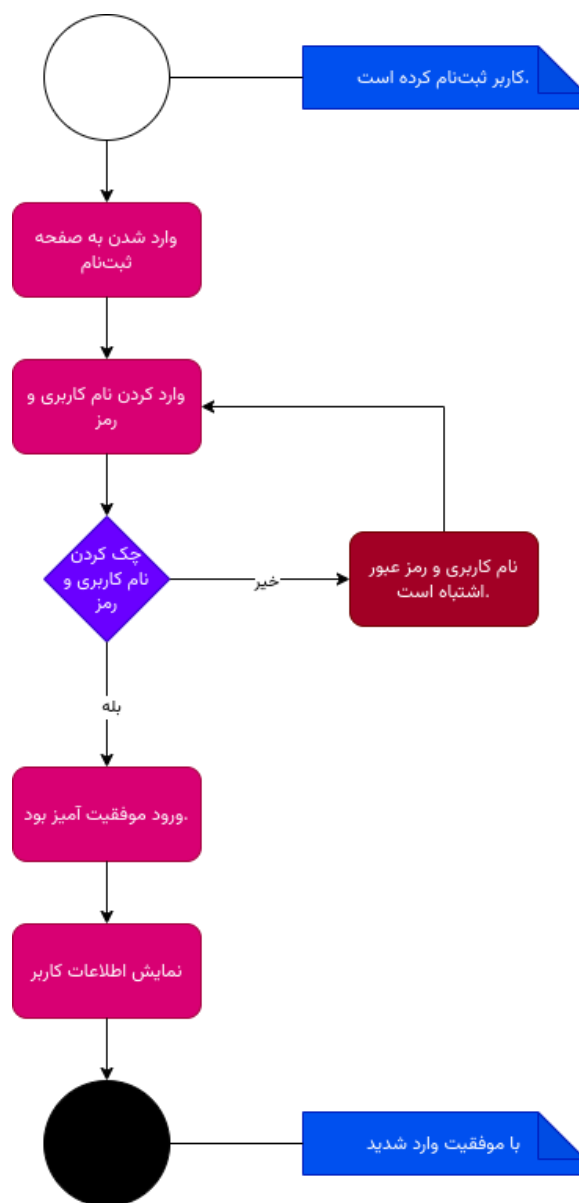
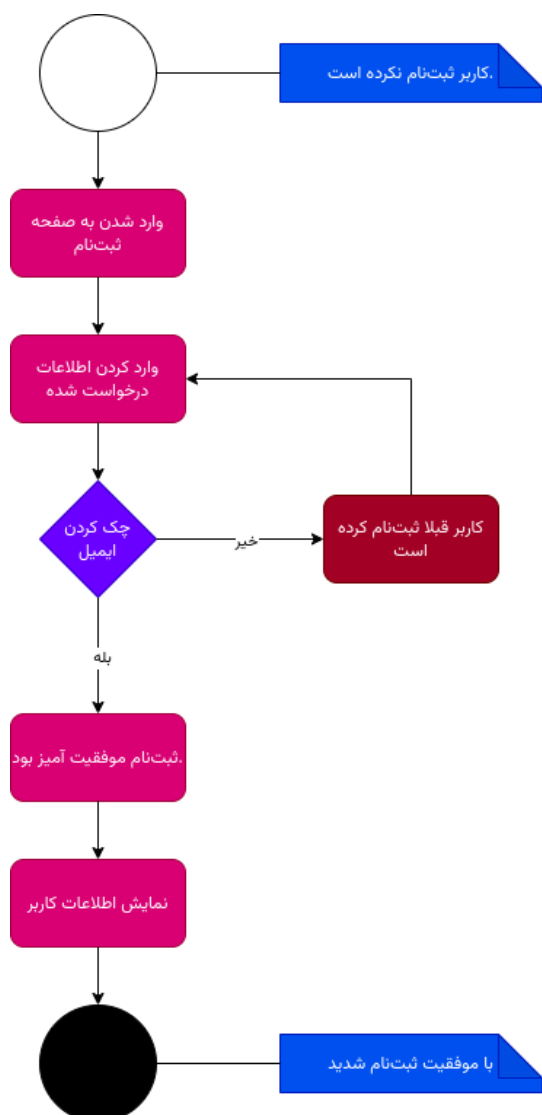
نمودار ۲- Use Case Diagram کاربر یا مشتری

## ۲.۴.۲. نمودار فعالیت

در این قسمت به نمایش نمودار فعالیت یا همان Activity Diagram می‌پردازیم و قسمت‌های مختلف وب‌سایت مانند ایجاد حساب کاربری و یا ورود به سایت و همچنین جزئیاتی که در نمودار Use Case Diagram مشاهده کردیم را به صورت ریزتر و مراحل انجام آنها را باهم مشاهده می‌نماییم.

## ۲.۴.۲.۱. ثبت نام

جهت ثبت نام جدید کاربر، در حالی که قبلا کاربر ثبت نام نکرده باشد، باید اول وارد صفحه ثبت نام شده، سپس اطلاعات درخواست شده توسط سیستم را وارد نمایید و بعد از آن روی دکمه ثبت نام کلیک کنید. در صورت نداشتن مشکلی، ثبت نام با موفقیت انجام خواهد شد.

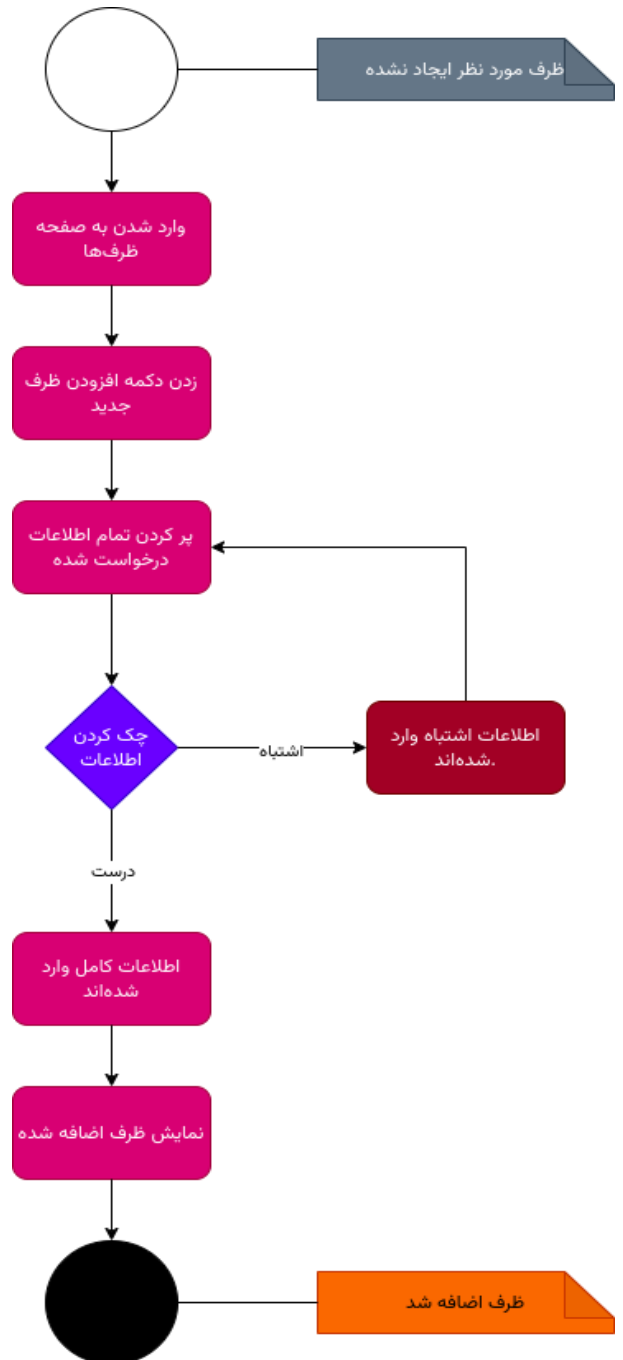
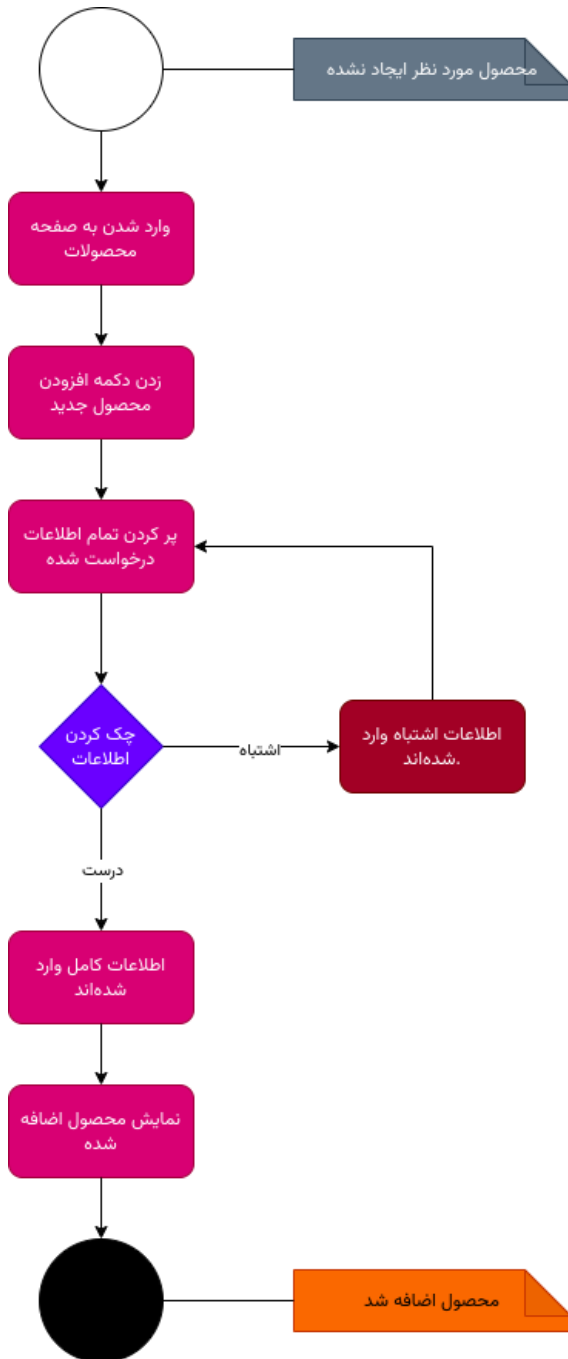


## ۲.۴.۲.۲. ورود

جهت ورود به سایت، در صورتی که قبلا ثبت نام کرده اید، باید وارد صفحه ثبت نام شده و سپس نام کاربری و رمز عبور خود را وارد کنید. در صورتی که اطلاعات وارد شده درست باشند و مشکلی نباشد، ورود با موفقیت انجام می شود.

### ۲.۴.۲.۳. افزودن محصول جدید

برای افزودن محصول جدید باید وارد بخش محصولات شده و سپس دکمه افزودن محصول جدید را بزنیم. سپس اطلاعات درخواست شده را وارد کرده و روی دکمه ثبت کلیک می‌کنیم.



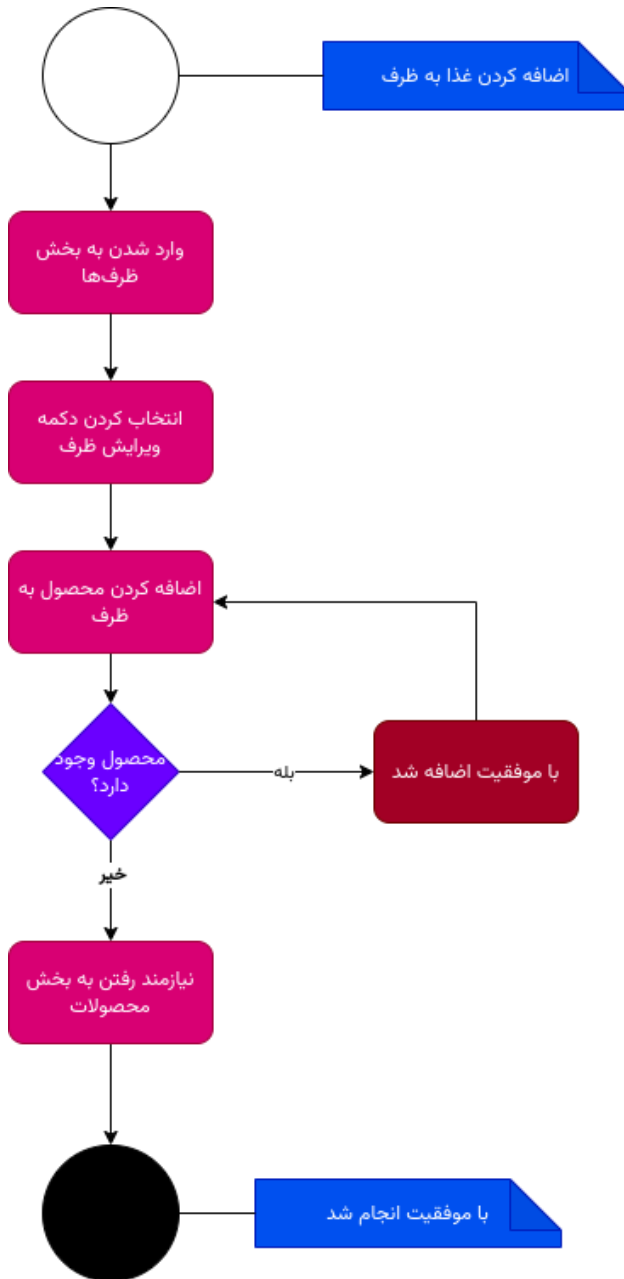
### ۲.۴.۲.۴. افزودن ظرف جدید

برای افزودن ظرف جدید به منوی غذایی خود، طبق مراحل ذکر شده جلو می‌رویم.



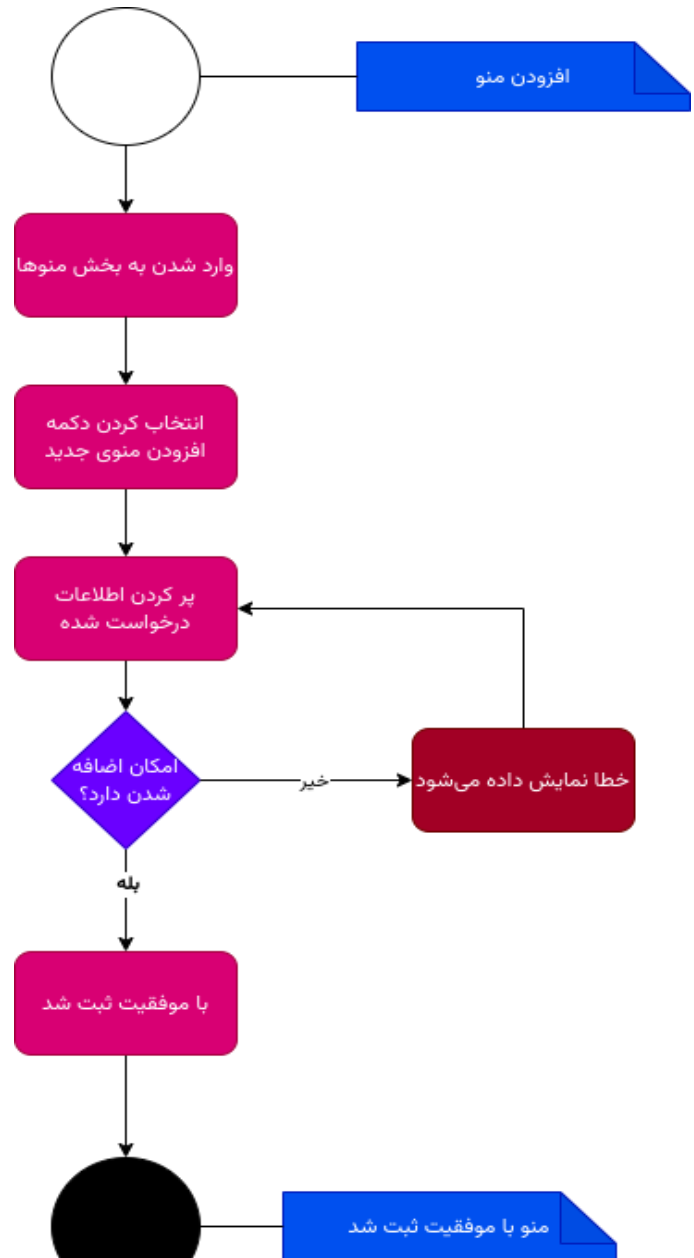
## ۲.۴.۲.۵. افزودن محصول به ظرف

جهت افزودن محصول به ظرف، مانند افزودن چای به ظرف صبحانه، باید به بخش ظرف‌ها وارد شویم و سپس ویرایش ظرف موجود را زده، محصول موردنظر را که از قبل در بخش محصولات وارد کرده‌ایم انتخاب می‌نماییم.



## ۲.۴.۲.۶. افزودن منوی جدید

برای افزودن منوی غذایی جدید به رژیم خود، باید طبق این دیاگرام مراحل را جلو ببرید. در اولین مرحله وارد بخش منوها می‌شویم، سپس روی دکمه افزودن منوی جدید کلیک می‌کنیم و اطلاعات درخواست شده را پر می‌کنیم و در نهایت روی ثبت کلیک می‌کنیم.



# فصل سوم

روش اجرای پروژه

## ۳.۱. احراز هویت، ورود به سیستم و سطح دسترسی‌ها

یکی از موارد بسیار مهم در تولید یک سیستم اتوماسیونی احراز هویت و مشخص کردن سطح دسترسی‌های هر کاربر از یک دیگر است به طوری که تا حد امکان مواردی که مورد نیاز کاربر ساخته شده توسط مدیر سامانه است در دسترس او باشند. برای سیستم‌هایی که نیازهای بسیار مشخص و اندکی دارند و کاربران آن نیز فقط موارد محدود تولید شده استفاده میکنند ایجاد سطح دسترسی‌های ثابت بسیار مرسوم است اما برای ایجاد انعطاف بیشتر و هموار کردن روند توسعه برای آینده سیستم تلاش کردم از سیستم ایجاد سطح دسترسی پویا استفاده کنم. این سیستم به ما کمک می‌کند که با تخصیص سطح دسترسی‌های گوناگون انعطاف بیشتری برای ایجاد نقش جدید در سیستم داشته باشیم.

## ۳.۲. تنظیمات اولیه

### ۳.۲.۱. نصب لاراول

برای نصب لاراول نیاز به یک‌سری ابزار داریم؛ لاراول از دو طریق کامپوزر و داکر قابل نصب می‌باشد که برای راحتی بیشتر از کامپوزر استفاده می‌کنیم و بعد از نصب کامپوزر در خط فرمان لینوکس و ویندوز به راحتی و با آموزشی که در داکيومنت خود سایت لاراول هست مراحل نصب را ادامه می‌دهیم و منتظر نصب لاراول می‌مانیم؛

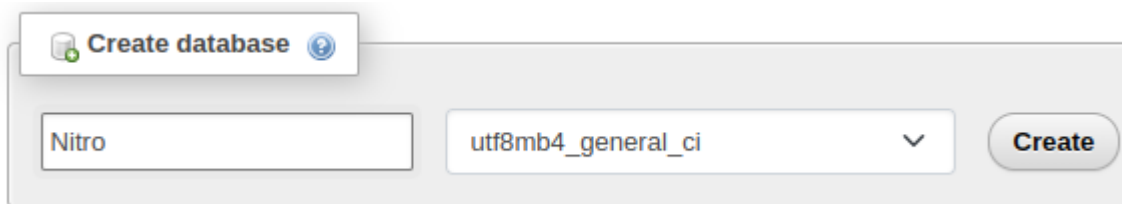
### ۳.۲.۲. نصب سرور مجازی

پس از نصب لاراول باید یک سرور مجازی مانند XAMPP یا LAMP را نصب و اجرا کنیم که این نرم‌افزار به ما کمک می‌کند تا تا MySQL و APACHE و متعلقات مورد نیاز یک سرور به راحتی روی کامپیوتر شخصی ما نصب شود.

### ۳.۲.۳. ساخت دیتابیس

باید برای ذخیره اطلاعات کاربران، محصولات و... یک دیتابیس داشته باشیم که تمام اطلاعات مورد نیاز در آن ذخیره شود. پس وارد Localhost/PhpMyAdmin می‌شویم و یک دیتابیس خالی ایجاد می‌نماییم.

برای ایجاد دیتابیس روی گزینه NEW زده و اطلاعات را به این صورت وارد می‌کنیم و سپس روی Create کلیک می‌کنیم.



### ۱۳.۲.۴ اجرای پروژه

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=Nitro
DB_USERNAME=root
DB_PASSWORD=
```

در نهایت باید با یک IDE مثل VSCode که رایگان است، پروژه را اجرا کنیم و به فایل ENV وارد شویم و در قسمت دیتابیس نام دیتابیس ساخته شده و پسورد (که روی لوکال به صورت پیش فرض خالی می باشد) را بنویسیم و صفحه را رفرش کنیم.

```
APP_NAME=NitroHealth
APP_ENV=local
APP_KEY=base64:Tm1eaZZjC01S3mUs5If
IsaXPzEoFxhnakZs0s80l9U=
APP_DEBUG=true
APP_URL=http://localhost
```

همچنین در بالای فایل ENV باید APP\_NAME را متناسب با نام پروژه خود قرار داده و APP\_URL را نیز آدرس سایت قرار داد؛ و اینکه پس از طراحی و برنامه نویسی پروژه باید APP\_DEBUG را False کرد تا ارورهای برنامه نویسی به کاربر نمایش داده نشوند.

در جدول زیر به وضوح تمام قسمت‌های دیتابیس قابل مشاهده هستند:

ولی لازم به ذکر است که نیازی به ساختن دستی هیچکدام از این جدول‌ها نیست و باید با استفاده از مایگريشن‌های لاراول استفاده کنیم.

### ۳.۳.۱. مایگریشن

مایگريشن‌ها به تیم مهندسان تولید نرم‌افزار کمک می‌کند تا تغییرات بانک اطلاعاتی را همانند سورس کد پروژه Commit کنند و اعضای تیم همیشه یک نسخه بروز از database را بر روی سیستم خود داشته باشند. مایگريشن‌ها بطور کلی از Schema Builder استفاده می‌کنند تا به راحتی بتوانند تغییرات را بر روی ساختار (Schema) بانک اطلاعاتی اعمال و مدیریت کنند.

### ۳.۳.۱.۱ ساخت Migration

برای ساخت یک Migration، دستور make:migration را در Artisan CLI وارد می‌کنیم:

```
php artisan make:migration create_users_table
```

سپس Migration در مسیر database/migrations تولید خواهد شد. فریم‌ورک برای نام‌گذاری فایل‌های Migration از یک Timestamp استفاده می‌نماید که ترتیب اجرای آنها را مشخص می‌کند. برای تعیین اینکه Migration بر روی کدام جدول اعمال شود، پارامتر --table را به آخر دستور آرטיسان اضافه می‌کنیم. و اگر Migration قرار است یک جدول جدید درون بانک اطلاعاتی ایجاد نماید، --create را به پارامترهای خود اضافه می‌کنیم:

```
php artisan make:migration add_votes_to_users_table --table=users
php artisan make:migration create_users_table --create=users
```

### ۳.۳.۱.۲ اجرای Migration

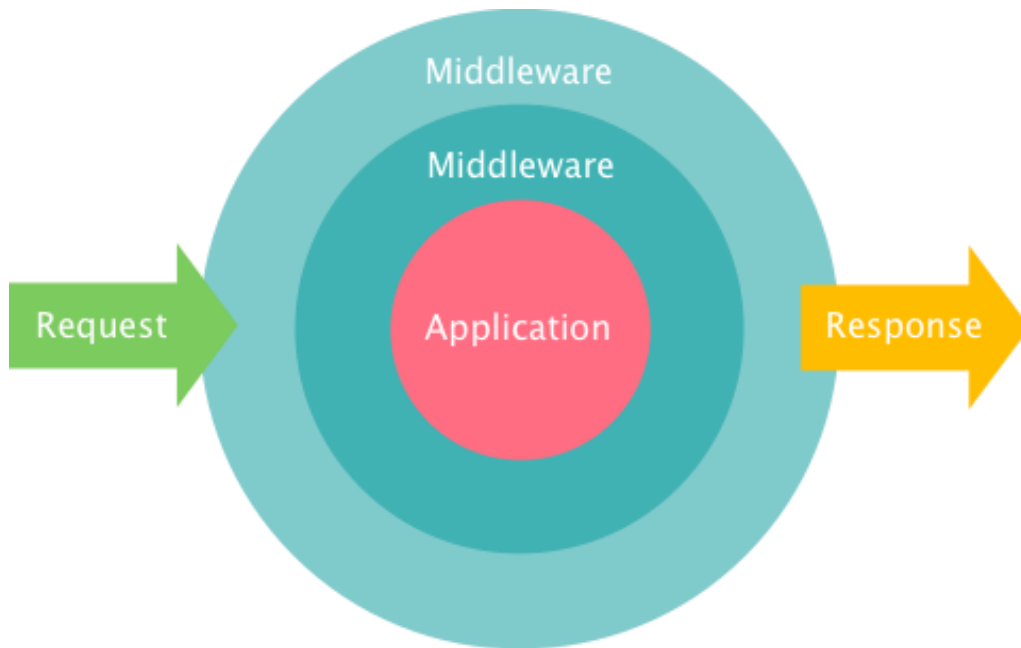
کد زیر تمامی Migrationها را بر روی دیتابیس اجرا می‌کند:

```
php artisan migrate
```

## ۳.۴. میان‌افزار (MiddleWare)

middleware در لغت به معنای "میان‌افزار" است و در واقع مانند پلی بین درخواست و پاسخ عمل می‌کند. به عبارتی دیگر middleware یک نوع مکانیزم فیلتر است که، می‌توان درخواست‌های HTTP را قبل از ورود به برنامه فیلتر کرد.

به عنوان مثال لاراول دارای یک middleware هست که می‌تواند مشخص کند کاربری که به برنامه وارد شده است احراز هویت شده است یا خیر. اگر کاربر مورد تایید نباشد، middleware، کاربر را به صفحه ورود به سایت یا صفحه Login هدایت می‌کند. و اگر کاربر از لحاظ هویتی تایید شده باشد، middleware به کاربر اجازه ورود می‌دهد و کاربر می‌تواند به برنامه وارد شود و درخواست‌هایش توسط سرور پردازش شود.



در لاراول middleware های زیادی وجود دارد، از جمله middleware : احراز هویت و حفاظت CSRF که در مسیر `app/http/Middleware` می‌توانید آنها را مشاهده نمایید.

## ۳.۵. قالب وبسایت

پس از انجام مایگريشن باید قالب وبسایت را وارد پروژه لاراولی خود کنیم؛ لازم به ذکر است که قالب وبسایت این پروژه با استفاده از فریم‌ورک محبوب Vue.js کدنویسی و طراحی شده است.

در ابتدا باید فایل‌های CSS و سپس فایل‌های Vue.js را وارد پروژه نمود و سپس در فایل WebPack آدرس آنها را بنویسیم تا به راحتی این فایل‌ها به فایل اصلی لاراول استخراج شوند.

## ۳.۶. لاراوول میکس

لاراوول میکس در اصل یه پکیج است که می‌شود خارج از پروژه‌های لاراوولی هم از آن استفاده کرد. ولی وقتی لاراوول نصب می‌شود، در آن این پکیج موجود است. این پکیج مستقیماً مربوط به فرانت‌اند (frontend) پروژه می‌شود و کار اصلی آن کامپایل کردن فایل‌های assets پروژه است. این ابزار شبیه Grunt, gulp و پکیج‌های مشابه عمل می‌کند. لاراوول میکس فایل‌های JS, LESS, SCSS, CSS و تمامی فایل‌های مربوط به assets را با هم کامپایل (ترکیب و فشرده) می‌کند و داخل فولدر public پروژه ذخیره می‌کند. نمونه ساده استفاده از آن به شکل زیر است:

```
mix.js('resources/js/app.js', 'public/js').vue({ version: 2 })
  .postCss('resources/css/app.css', 'public/css', [
    //
  ]);
```

پیش نیاز لاراوول میکس NPM است و نیاز است که حتماً نصب شود؛ در اوبونتو به راحتی با نوشتن NPM Install در ترمینال نصب می‌شود. پس از نصب NPM نیز باید دستور NPM Run Dev یا NPM Run Watch را اجرا کرد تا کامپایل انجام شود.

تفاوت اصلی این دو دستور در این است که NPM Run Dev فقط یک‌بار کامپایل را انجام می‌دهد ولی NPM Run Watch حواسش را به کد می‌دهد و با ایجاد هر تغییر یک‌بار دیگر کامپایل انجام می‌شود.

## ۳.۷. ایجاد محصولات

در ابتدا باید مدل و کنترلر این بخش با استفاده از دستورات:

ساخت و سپس مدل را به دیتابیس متصل کرد و در بخش‌های مختلف کنترلر دستورات مورد نیاز را نوشت.

```
public function index()
{
    $products = ProductsResource::collection(Products::with('products_categories')->get());
    return $products;
}
```



در بخش Store در کنترلر محصولات باید تمام اطلاعاتی که برای دیتابیس نیاز داریم را بسازیم و اطلاعاتی که در پنل وارد می‌شوند را Validate یا تایید کنیم و ببینیم که مقادیر درست هستند یا خیر.

```
public function store(Request $request)
{
    $data = $request->form;
    $data['products_subcategories_id'] = 1;
    $data['sort'] = 0;
    $new_products = Products::create($data);
    return new ProductsResource($new_products);
}
```

همانطور که در بالا اشاره شد، در فایل مدل باید جدول مربوط به مدل را معرفی کرد و نوع اتصال به دیتابیس را نیز تعیین نمود.

```
ProductsResource.php

public function toArray($request)
{
    //return parent::toArray($request);
    return [
        'id' => $this->id,
        'name' => $this->name,
        'products_categories_id' => $this->products_categories_id,
        'protein' => $this->protein,
        'fat' => $this->fat,
        'carbohydrates_total' => $this->carbohydrates_total,
        'vitamin_a' => $this->vitamin_a,
        'vitamin_b1' => $this->vitamin_b1,
        'vitamin_b2' => $this->vitamin_b2,
        'created_at' => date('d.m.Y', strtotime($this->created_at)),
        'products_categories' => new ProductsCategoryResource($this->products_categories),
    ];
}
```

تمام اطلاعات ستون‌های دیتابیس در این قسمت نوشته می‌شوند.

برای بخش View نیز در چند قسمت کد نوشته شده است مانند این قسمت که دارای یک فرم برای نمایش محصولات می‌باشد:

```
Products.Vue

<template>
  <div style="width:90%; margin: 0 auto;">
    <FlashMessage :position="'right top'"></FlashMessage>
    <h1 class="text-center">لیست محصولات</h1>
    <b-button v-b-modal.modal-1 class="mb-4 main-button-green">افزودن محصول جدید</b-button>
    <b-modal id="modal-1" class="mb-4" size="lg" title="افزودن محصول جدید" hide-
      footer="true">
      <addProduct />
    </b-modal>
    <table class="fl-table">
      <thead>
        <tr>
          <th scope="col">ترتیب</th>
          <th scope="col">نام</th>
          <th scope="col">دسته بندی</th>
          <th scope="col">پروتئین</th>
          <th scope="col">چربی</th>
          <th scope="col">کربوهیدرات</th>
          <th scope="col">ویتا مین A</th>
          <th scope="col">ویتا مین B1</th>
          <th scope="col">ویتا مین B2</th>
          <th scope="col">کنترل</th>
        </tr>
      </thead>
      <tbody>
        <tr v-if="allProducts.length === 0 && getLoading === false">
          <td class="text-center text-danger" colspan="10"><b>! داده یافت نشد</b></td>
        </tr>
        <tr v-else-if="getLoading === true" style="border:none">
          <td class="text-center text-danger" colspan="10">
            <Loader />
          </td>
        </tr>
        <tr v-else v-for="(myProduct, index) in allProducts" :key="index">
          <td>{{myProduct.id}}</td>
          <td>{{myProduct.name}}</td>
          <td>{{myProduct.products_categories.name}}</td>
          <td>{{myProduct.protein}}</td>
          <td>{{myProduct.fat}}</td>
          <td>{{myProduct.carbohydrates_total}}</td>
          <td>{{myProduct.vitamin_a}}</td>
          <td>{{myProduct.vitamin_b1}}</td>
          <td>{{myProduct.vitamin_b2}}</td>
          <td class="text-center">
            <b-button variant="danger" @click="removeProducts(myProduct.id)">
              <font-awesome-icon icon="trash" />
            </b-button>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
</template>
```

## ۳.۸. مسیرها (Route ها)

در لاراول درخواست‌های فرستاده شده از URL را قسمت‌های خاصی از برنامه بررسی می‌کند. اینکه کدام درخواست در کدام قسمت از برنامه باید بررسی بشود را روترها مشخص می‌کنند. در واقع مسیریابی درخواست‌ها را روترها انجام می‌دهند. اگر درخواستی داشته باشیم که مسیری برای آن وجود نداشته باشد با خطای ۴۰۴ مواجه می‌شویم.

تمامی روت‌های یک پروژه در فایل‌های web.php و api.php در پوشه routes قرار دارند. برای روت‌های فایل web.php به صورت خودکار، میدلورهای session و csrf اعمال می‌شوند. این میدلورها در لاراول، در روت‌های فایل api.php اعمال نمی‌شوند، به خاطر اینکه روت apiها در این فایل مشخص می‌شوند و درخواست‌هایی که با apiها انجام می‌شوند، نیازی به این میدلورها ندارند.

```
api.php

Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});
Route::middleware('auth:sanctum')->get('/authenticated', function () {
    return true;
});
```

آدرس‌های مربوط به میدلورها

```
api.php

Route::prefix('/v1/dishes')->group(function(){
    Route::get('/index', [DishesController::class, 'index']);
    Route::post('/store', [DishesController::class, 'store']);
    Route::delete('/{id}', [DishesController::class, 'destroy']);
    Route::get('/dishes-categories', [DishesController::class, 'dishesCategories']);
    Route::get('/culinary-processings', [DishesController::class, 'culinaryProcessings']);
    Route::post('/copy-dish/{id}', [DishesController::class, 'copyDish']);
    Route::post('/one-dish/{id}', [DishesController::class, 'oneDish']);
});
```

آدرس‌های مربوط به بخش ظرف‌ها

```

api.php

Route::prefix('v1')->group(function(){
    Route::get('/recipes-collections', [RecipesCollectionsController::class, '__invoke']);
});

```

آدرس مربوط به بخش دستورات غذایی

```

api.php

Route::prefix('v1/menus')->group(function(){
    Route::get('/', [MenuController::class, 'index']);
    Route::get('/first-menu', [MenuController::class, 'firstMenu']);
    Route::post('/store', [MenuController::class, 'store']);
    Route::delete('/{id}', [MenuController::class, 'destroy']);
    Route::get('/menu-characters/{id}', [MenuController::class, 'menuCharacters']);
});

```

آدرس‌های مربوط به منوی غذایی

```

api.php

Route::prefix('v1/dishes-products')->group(function(){
    Route::post('/store', [DishesProductsController::class, 'store']);
    Route::delete('/{id}', [DishesProductsController::class, 'destroy']);
    Route::post('/products-dish/{id}', [DishesProductsController::class, 'productsDish']);
    Route::delete('/{id}', [DishesProductsController::class, 'destroy']);
});

```

آدرس‌های مربوط به بخش محصولات در ظرف

```

api.php

Route::prefix('v1/products')->group(function(){
    Route::get('/', [ProductsController::class, 'index']);
    Route::post('/store', [ProductsController::class, 'store'])->middleware('auth:sanctum');
    Route::get('/products-category', [ProductsController::class, 'productsCategory']);
    Route::delete('/{id}', [ProductsController::class, 'destroy']);
});

```

آدرس‌های مربوط به بخش محصولات



api.php

```
Route::prefix('v1/register')->group(function(){
    Route::get('/roles', [RegisterController::class, 'roles']);
    Route::get('/headerlinks', [RegisterController::class, 'headerlinks']);
});
```

آدرس‌های مربوط به ثبت‌نام

## ۳.۹. بخش احراز هویت

```
LoginController.php

public function login(Request $request)
{
    $request->validate([
        'email' => 'required|email',
        'password' => 'required',
    ]);
    if (Auth::attempt($request->only('email', 'password')))
    {
        return response()->json(Auth::guard('web')->user(), 200);
    }
    throw ValidationException::withMessages([
        'email' => ['نام کاربری یا رمز اشتباه است']
    ]);
}
```

تکه کد مربوط به کنترلر بخش ورود

```
Register.vue

<script>

export default {
  data:function(){
    return{
      form:{
        name:'',
        email:'',
        password:'',
        password_confirmation:''
      },
      errors:[]
    }
  },
  methods:{
    saveForm(){
      console.log('send')
      axios.post('/register', this.form)
        .then(prevent =>{
          console.log('saved')
          this.$router.push({ name: 'login' })
        })
        .catch(error => {

        })
    }
  }
}

</script>
```

تکه کد View مربوط به بخش ثبت نام

```

Login.Vue

<script>
import {email, required, minLength} from 'vuelidate/lib/validators'
import {mapGetters, mapActions} from 'vuex'

export default{

  data:function(){
    return{
      form:{
        email:'',
        password:'',
      },
      errors:null,
    }
  },
  validations: {
    form:{
      email: {email, required},
      password: {required, minLength: minLength(6)}
    }
  },
  methods:{
    login(){
      if (this.$v.$invalid) {
        this.$v.$touch();
        return
      }
      axios.get('/sanctum/csrf-cookie').then(response => {
        axios.post('/login', this.form)
        .then(response =>{
          localStorage.setItem('x_xsrftoken', response.config.headers['X-XSRF-
TOKEN']);
          this.$router.push({name:'cabinet'});
        })
        .catch(response => {
          //if (error.response.status === 422) {
            this.errors = 'احتمالاً نام کاربری یا رمز عبور اشتباه است';
          //}
        });
      })
    }
  }
}
</script>

```

تکه کد مربوط به بخش View ثبت نام

```

Dish.vue

<template>
  <tr><td scope="row" class="text-center">{{dish.id}}</td>
    <td>{{dish.name}}</td>
    <td class="text-center">{{dish.dishes_categories.name}}</td>
    <td class="text-center">{{dish.recipes_collections.name}}</td>
    <td class="text-center">{{dish.yield}}</td>
    <td class="text-center">{{dish.techmap_number}}</td>
    <td class="text-center"><b-button variant="danger" @click="removeDish()">
      <font-awesome-icon icon="trash"/>
    </b-button></td>
    <td class="text-center"><b-button variant="success" @click="$emit('addProductsDish', dish.id)">
      <font-awesome-icon icon="plus-square"/>
    </b-button></td>
    <td class="text-center"><b-button @click="copyDishClick()">
      <font-awesome-icon icon="copy"/>
    </b-button></td>
  </tr>
</template>
<script>
import {mapGetters, mapActions} from 'vuex'
export default{
  props:['dish'],
  computed: mapGetters(['allDishesCategories', 'myRecipesCollections', 'allCulinaryProcessings']),
  methods:{
    ... mapActions(['deleteDish', 'copyDish']),
    removeDish(){
      this.deleteDish(this.dish.id)
      .then(response => {
        this.flashMessage.success({
          message: 'طرف با موفقیت حذف شد',
          time: 3000,
        });
      });
    },
    copyDishClick(){
      this.copyDish(this.dish.id);
    },
  },
}
</script>

```

کد مربوط به بخش نمایش ظرف

```

Authenticate.php

protected function redirectTo($request)
{
  if (! $request->expectsJson()) {
    return route('login');
  }
}

```

تکه کد مربوط به *Middleware* بخش احراز هویت



# فصل چهارم

تجزیه و تحلیل داده‌ها (توضیح رابط کاربری و نحوه کارکرد سیستم)

پروژه نیتروی سلامتی، یک پروژه سلامتی است که در حال حاضر و در فاز اول شامل کالری شمار است که از سمت کاربر و ادمین قابلیت افزودن محصولات، ظروف غذایی، رژیم‌ها و منوهای غذایی امکان‌پذیر می‌باشد. همچنین در این فاز، نمایش گزارش کالری مصرفی و... نمایش داده می‌شود.

لازم به ذکر است که در فصل دوم دیتابیس و نحوه کار به طور کامل شرح گردیده است و از گزافه‌گویی در این قسمت خودداری شده است و فقط به توضیح مختصر و تصویر هر بخش بسنده شده است.

## ۴.۱. صفحات اصلی سایت

ثبت‌نام

ورود

NitroHealth

---

ورود

ایمیل

user@example.com

رمز

ورود

ایمیل: Info@ArsalanArghavan.ir

رمز: 12345678



### عکس ۴.۱. ورود به سایت

این قسمت از وب‌سایت مربوط به ورود به سایت می‌باشد که پس از ثبت‌نام می‌توانیم از این قسمت وارد حساب کاربری خود که شامل مدیر و یا کاربر می‌باشد شویم. با توجه به کاربرد و هویت سایت، برای استفاده از امکانات سایت و ثبت اطلاعات در دیتابیس، نیاز است که به حساب کاربری خود وارد شویم. پس ورود به وب‌سایت برای استفاده از امکانات آن الزامی می‌باشد.

## ثبت نام

نام

ایمیل

رمز

تکرار رمز

ثبت نام



## عکس ۴.۲. ثبت نام

همانطور که در بخش قبلی ذکر شد، ورود به سایت برای استفاده از امکانات سایت الزامی می باشد. پس قبل از ورود به وبسایت نیازمند ثبت نام هستیم. ثبت نام در وبسایت در فاز اول فقط شامل نام، ایمیل، رمز و تکرار رمز می باشد.

نام کاربری: Arsalan

ایمیل: info@arsalanarghavan.ir

نقش کاربری:

خروج



## عکس ۴.۳. داشبورد کاربری

بعد از ورود به وبسایت وارد داشبورد کاربری می شویم. در فاز اول پروژه برای داشبورد امکاناتی در نظر گرفته نشده و صرفاً به نمایش نام کاربری وارد شده بسنده شده است.

## لیست محصولات

افزودن محصول جدید

ترتیب	نام	دسته‌بندی	پروتئین	چربی	کربوهیدرات	ویتامین A	ویتامین B1	ویتامین B2	کنترل
26	مربا هویج مجید	شیرینیجات	50	60	20	20	15	18	
27	سویا	حیوانات	10	65	80	55	45	4	
31	لوریا فرنگی	صیفی‌جات	85	65	18	55	44	46	



## عکس ۴.۴. لیست محصولات

در این قسمت از وب‌سایت، لیست محصولات قابل مشاهده است. محصولات، تمام خوراکی‌هایی هستند که دارای کالری، ویتامین، پروتئین، چربی و مواد معدنی هستند. از این قسمت می‌توانیم محصولات را حذف کنیم و یا از قسمت افزودن محصول جدید، محصول خود را اضافه نماییم.

افزودن محصول جدید

نام محصول

دسته‌بندی محصولات

آب

پروتئین

چربی

کربوهیدرات، گرم

رئز عناصر

سدیم، میلی گرم

پتاسیم، میلی گرم

کلسیم، میلی گرم

هنیزیم، میلی گرم

فسفر، میلی گرم

آهن، میلی گرم

ید، میلی گرم

سلنیوم، میلی گرم

ویتامین‌ها

ویتامین A، میلی گرم

ویتامین B1، میلی گرم

ویتامین B2، میلی گرم

ویتامین C، میلی گرم

ویتامین D، میلی گرم

ذخیره محصول

افزودن محصول جدید

ترتیب	نام
26	مربا هویج مجید
27	سویا
31	لوریا فرنگی

## عکس ۴.۵. افزودن محصول جدید

از این قسمت می‌توانیم محصولات موردنیاز خود را اضافه نماییم. برای افزودن محصول نیاز است که اطلاعات تغذیه‌ای محصول را وارد نماییم.

NitroHealth								
محصولات   طرف‌ها   منوی غذایی   غذاها   گزارش   حساب کاربری								
لیست ظرف‌ها								
ایجاد ظرف جدید								
ترتیب	نام	دسته‌بندی	مجموعه	خروجی	شماره	کنترل		
30	صبحانه	صبحانه	مجموعه هن	15	15			



#### ۴.۶. لیست ظرف‌ها

در این قسمت از وبسایت، می‌توانیم ظرف‌های غذایی را مشاهده، کپی و یا حذف نماییم. ظرف‌های غذایی می‌توانند دارای ویژگی، چند غذای مختلف، دستور پخت و... باشند.

ایجاد ظرف جدید

نام ظرف

دسته‌بندی ظرف‌ها

دستورهای پخت

فرآیند پخت و پز

ویژگی‌های غذا

شماره

عملکرد بر حسب گرم

تکنولوژی پخت و پز

دخیره غذا

#### ۴.۷. عکس افزودن ظرف جدید

در این قسمت می‌توانیم ظرف مورد نیاز خود را اضافه نماییم. اضافه کردن ظرف منوط به پر کردن این اطلاعات است. همچنین بعد از افزودن ظرف، باید غذا را به ظرف اضافه نماییم. برای مثال یک ظرف به نام صبحانه وارد می‌کنیم و پس از آن، محصولاتمانند چای، کره، پنیر و... را به آن اضافه می‌نماییم.



### عکس ۴.۸. فهرست منو

در این قسمت تمام منوهای غذایی را می‌توانیم مشاهده نماییم. منوهای غذایی شامل نام منو، روزهای منو و وعده‌هایی که می‌توانیم این غذا را مصرف کنیم می‌باشند.



## عکس ۴.۹. منوی جدید

از این قسمت، می‌توانیم منوهای جدیدی ایجاد کنیم که باعث کمک به این می‌شود که بدانیم در چه روزی چه غذایی را می‌توانیم نوش جان کنیم!

[محصولات](#) [ظرف‌ها](#) [منوی غذایی](#) [غذاها](#) [گزارش](#) [حساب کاربری](#) **NitroHealth**

---

**عملکرد منو**

منو

روزها

وعده‌های غذایی

نمایش



## عکس ۴.۱۰. عملکرد منو

در این قسمت می‌توانیم تاریخچه تمام منوهای تعریف شده از قبل، وعده‌ها و... را جستجو کنیم. برای جستجو باید یک منو، یا روز و وعده را جستجو کرد.

[محصولات](#) [ظرف‌ها](#) [منوی غذایی](#) [غذاها](#) [گزارش](#) [حساب کاربری](#) **NitroHealth**

---

**گزارش منو**

منو

نمایش



## عکس ۴.۱۱. گزارشات

در این قسمت می‌توانیم تمام گزارشات عملکرد مربوط به منو، مانند کالری‌های مصرفی، غذاها، پروتئین و... را مشاهده کنیم.