

وب اپلیکیشن اپکیت

فایل راهنما

فهرست مطالب

روی هر یک از عناوین کلیک کنید تا به بخش مورد نظر هدایت شوید

- معرفی
- ساختار قالب
- کلاس های کمکی
- هایلایت و پس زمینه ها
- منو
- منو اصلی
- وب اپلیکیشن
- صفحات و المان ها

خوش آمدید

در این راهنما، ما تمام جنبه های مورد نیاز قالب را پوشش خواهیم داد تا به شما در دستیابی به بهترین وب اپلیکیشن (PWA) کمک کنیم. اپکیت توسط بوت استرپ ۵، جاوا اسکریپت و بدون وابستگی های جی کوئری، طراحی شده است. همه المان های بوت استرپ کاملاً با اپکیت سازگار خواهند بود و همچنین المان های ساخته شده سفارشی ما استفاده و درک آن برای شما بسیار آسان خواهد بود.

توجه کنید که این راهنما به احتمال زیاد کاملاً هر سؤالی را که ممکن است داشته باشید پوشش می دهد. ما این راهنما را نوشته ایم تا هم برای توسعه دهندگان و هم کاربران با تجربه کمتر، مناسب باشد.

هدف ما ارائه بهترین تجربه ممکن به شما است، بنابراین از اگر هر گونه سوالی دارید از طریق تیکت در راست چین در ارتباط باشید.

لطفاً برای مطالعه این راهنما وقت بگذارید. بعد از مطالعه این راهنما، همه چیز فقط کپی و پیست خواهد بود. ما تمام تلاش خود را کرده‌ایم تا کدنویسی قالب را تا حد امکان کلاسیک برای توسعه‌دهندگان نگه داریم و برای کاربرانی که تجربه کمتری دارند قابل درک باشد.

درک بهتر این راهنما

کل سیستم قالب بر اساس المان ساخته شده است، همچنین مقدار المان مورد نیاز برای ایجاد یک صفحه با کدنویسی آنها به سادگی راه حل بهینه نیست. برای درک کامل محصول، اکیداً توصیه می‌کنیم راهنما را به ترتیب زیر مطالعه کنید.

ساختار کلی قالب ([اینجا](#))

راهنمای المان ها ([اینجا](#))

راهنمای صفحات ([اینجا](#))

سوالات متداول ([اینجا](#))

با درک ساختار اصلی و کلاس های کاربردی و سپس اجزای سازنده، به سرعت فوق العاده ای قالب را درک خواهید کرد! المان ها ساخت صفحات جدید را آسان و فوق‌العاده سریع میکند و در هنگام افزودن ایده‌های خود به محصول ما، انعطاف‌پذیری بی‌سابقه‌ای را به شما ارائه می‌دهد.

ساختار کلی قالب

توضیحات صفحه [start_sample.html](#) و ساختار کلی صفحات

ساختار اصلی قالب در صفحه [start_sample.html](#) نشان داده شده است. در اینجا می‌توانید آن را با المان یا عناصر صفحه، المان های بوت استرپ در صورت نیاز پر کنید، اما، برای جلوگیری از خطا، باید مطمئن شویم که آن را درک کرده‌اید.

ساختار صفحه

```

1. <!-- Preloader will be added here-->
2.
3. <div id="page">
4.
5.     <!-- Header Bar will be added here -->
6.     <!-- Footer Bar will be added here -->
7.
8.     <div class="page-content header-clear-VALUE">
9.
10.         <!-- Add your Canvas Elements - meanign - Components and Page Elements Here-->
11.
12.     </div>
13.
14.     <!-- Off Canvas Elements Here -->
15.
16. </div>
17.

```

این ساختار اصلی قالب است. بنابراین پیروی از این ساختار برای عملکرد بهتر قالب ضروری است، به خصوص که ما از بسیاری از عناصر خارج از بوم برای ایجاد یک تجربه کاربری زیبا استفاده می کنیم.

خط ۱ – المان لودر صفحه است. فقط این المان خارج از صفحه خواهد بود.

خط ۳ – تگ صفحه اصلی است. هر عنصر در صفحه خود را در اینجا نگه دارید.

خط ۵ – جایی است که **نوار هدر** قرار میگیرد.

خط ۶ – جایی است که **نوار فوتر** قرار میگیرد.

خط ۸ – محتوای صفحه است. همه المان های درون صفحه ای را شامل میشود(همه عناصر، به استثنای چند مورد که در زیر توضیح داده شده است)

خط ۱۲ – اتمام محتوای صفحه.

خط ۱۴ – جایی است که ما عناصر خارج از بوم خود را در آن قرار می دهیم (منوها، مدال ها، اسنک بارها).

خط ۱۶ – تگ صفحه را می بندد و از این قسمت به بعد لطفاً مطالب بیشتری اضافه نکنید.

نکته: کلاس **page-content** از بالای صفحه شروع می شود، بنابراین وقتی محتوایی را در آن اضافه می کنید ممکن است توسط هدر پنهان شود. برای حل این مشکل، بسته به نیازتان، کلاس **header-clear** / **header-clear-small** / **header-clear-medium** / **header-clear-large** را به جای **header-clear-VALUE** به محتوای صفحه اضافه کنید.

المان های درون بومی چیست؟ (page-content)

بوم مکانی در نظر گرفته می شود که در آن همه عناصری را که در صفحه می بینید، که فقط هنگام اسکرول حرکت نمی کنند. هر چیزی که در کلاس **page-content** می بینید یک عنصر روی بوم در نظر گرفته می شود. این عناصر همیشه روی صفحه قابل مشاهده هستند و به جز زمانی که در حال پیمایش هستید حرکت نمی کنند.

عناصر خارج از بوم چیست؟ (محدوده بین page-content و صفحه)

این عناصر به عنوان عناصری تعریف می شوند که از سمت چپ، راست، بالا یا پایین صفحه ظاهر می شوند و از خارج از ناحیه قابل مشاهده صفحه می آیند. این عناصر در **خارج از محتوای صفحه** قرار می گیرند و این بسیار مهم است، زیرا آنها در تعامل هستند. با محتوای صفحه به عنوان مثال، منوها می توانند محتوای صفحه را فشار دهند و یک پوشش تاریک ایجاد کنند. به همین دلیل، لیست عناصر زیر همیشه باید خارج از محتوای صفحه قرار گیرد

- Header Bar
- Footer Bar
- Action Sheets
- Action Modals
- Sidebar Menus
- Toasts
- Snackbars

⚠ این عناصر را در کلاس page-content قرار ندهید. انجام این کار منجر به مشکلات z-index می شود، به این معنی که آنها به درستی نشان داده نمی شوند.

تگ body

تگ body برای کنترل تم صفحه و همچنین رنگ هایلایت و پس زمینه استفاده می شود.

```
1. <body class="theme-light">
```

کلاسی که برای تعیین تم رنگی محصول خود استفاده می کنیم. برای تشخیص خودکار تم پیش فرض دستگاه هایتان، از **theme-dark** برای حالت تاریک، **theme-light** برای حالت روشن و از **detect-theme** برای تشخیص خودکار استفاده کنید. اگر این مورد توسط دستگاه شما پشتیبانی نمی شود، صفحه به طور خودکار به صورت پیش فرض روی **theme-light** قرار می گیرد. اکثر دستگاه های مدرن اکنون از تشخیص خودکار تم پشتیبانی می کنند و می توانید این را در صفح [component-auto-dark.html](#) بررسی کنید.

هدر قالب به گونه ای طراحی شده است که استفاده از آن بسیار آسان است و برای موارد استفاده شما بسیار انعطاف پذیر است. با این گفته، هدر در حال حاضر از چند سبک پشتیبانی می کند که در زیر توضیح خواهیم داد و می توان آنها را در صفحه [component-headers.html](#) مشاهده و استفاده کرد.

```
1. <div class="header header-fixed header-logo-left">
2.   <a href="index.html" class="header-title">AppKit Mobile</a>
3.   <a href="#" class="header-icon header-icon-1"><i class="fas fa-envelope"></i><span class="badge bg green1-dark">5</span></a>
4.   <a href="#" class="header-icon header-icon-2"><i class="fas fa-bars"></i></a>
5.   <a href="#" class="header-icon header-icon-3"><i class="fas fa-heart color-red2-light"></i></a>
6.   <a href="#" class="header-icon header-icon-4"><i class="fas fa-share-alt"></i></a>
7. </div>
```

بیاید هدر را خط به خط توضیح دهیم و نشان دهیم که چه کاری می تواند انجام دهد.

خط ۱ کلاس بسته اصلی است. این شامل کلاس **header**، برای تعریف عنصر و **header-fixed** است، که باعث می شود هدر در بالای صفحه ثابت بماند و همچنین کلاسی که برای تعیین استایل استفاده می شود. می توانید از هر یک از کلاس های زیر استفاده کنید.

header-logo-left - لوگوی سمت چپ با ۴ آیکون در سمت راست
header-logo-right - لوگوی سمت راست با ۴ آیکون در سمت چپ
header-logo-center - لوگوی وسط با ۲ آیکون در راست و ۲ آیکون در چپ
header-logo-app - ۱ آیکون در سمت چپ، لوگو و ۳ آیکون در سمت راست.

خط ۲ لوگوی هدر است. می تواند یک تصویر یا یک متن باشد. با تنظیم کلاس **header-title** روی **header-logo** می توانید این را به عنوان تصویر انتخاب کنید. تصویر را می توان با جستجوی "**header-logo**" در فایل اصلی **css** پیدا کرد و آن را با تصویر خود جایگزین کرد.

خطوط ۳، ۴، ۵، ۶ آیکون های هدر هستند. با استفاده از **header-icon-۱** تا ۴ شماره گذاری می شوند و می توانید موقعیت آنها را بر اساس این عدد تغییر دهید.

فوتر

منوی فوتر چیزی است که به قالب جان بخشیده است. تعداد زیادی از مشتریان از ما خواسته اند تا منویی ایجاد کنیم که انعطاف پذیر، آسان برای استفاده باشد اما در عین حال چندین سبک زیبا را ارائه دهد. برای مشاهده تمام نوارهای فوتر موجود، لطفاً صفحه [component-footers.html](#) را بررسی کنید، جایی که می توانید نمونه های زنده را ببینید و می توانید آنها را مستقیماً در صفحات خود کپی کنید.

```
1. <div id="footer-bar" class="footer-bar-1">
2.   <a href="#"><i class="fa fa-home"></i><span>Home</span></a>
3.   <a href="#" class="active-nav"><i class="fa fa-star"></i><span>Features</span></a>
4.   <a href="#"><i class="fa fa-heart"></i><span>Pages</span></a>
5.   <a href="#"><i class="fa fa-search"></i><span>Search</span></a>
6.   <a href="#"><i class="fa fa-cog"></i><span>Settings</span></a>
7. </div>
```

منوی فوتر ما محدود به تعداد آیتم های شما در آن نیست. فوتر با استفاده از Flexbox ساخته شده است، بنابراین به طور خودکار همه موارد را به تنهایی محاسبه و تراز می کند. با این حال لطفاً توجه داشته باشید که دستگاه هایی با صفحه نمایش کوچک وجود دارند، بنابراین استفاده از بیش از ۵ آیکون توصیه نمی شود. در داخل id نوار فوتر، لینک ها و آیکون های ۶ FontAwesome خواهید داشت. در صورت نیاز آنها را ویرایش یا تغییر دهید. برای برجسته کردن/فعال کردن صفحه کافی است **active-nav** را به منوی فوتری که می خواهید اضافه کنید.

نتیجه گیری

ساختار صفحه در نظر گرفته شده است تا با استفاده از صفحه **_starter.html** تا حد امکان ساده باشد، عناصر بالا را در داخل آن پیدا خواهید کرد و می توانید بلافاصله محتوای خود را در کلاس محتوای صفحه قرار دهید! برای آشنایی با منوها و کلاس های رنگ، به خواندن راهنما ادامه دهید.

کلاس های کمکی

مهمترین ویژگی ها که به شما این امکان را می دهد تا بی نهایت به قالب سبک دهید

کلاس های ثانویه قطعات قابل استفاده مجدد CSS هستند که به HTML شما اضافه می شوند تا ویژگی های خاصی را تقویت و گسترش دهند. به عنوان مثال، ممکن است بخواهید دوباره از یک رنگ استفاده کنید، بنابراین به جای اینکه برای هر بار که آن رنگ را می خواهیم دوباره در CSS بنویسیم، یک کلاس برای آن ایجاد می کنیم. اینها کلاس های Bootstrap پیش فرض هستند که با چارچوب Bootstrap در دسترس هستند

getbootstrap.com/docs/5.3/

پس زمینه های قابل استفاده مجدد

ممکن است در کدهای قالب متوجه شده باشید که ما از bg-NUMBER برای پس زمینه استفاده می کنیم. اینها پس زمینه های نمایشی هستند که در فایل zDemo.scss یا داخل style.css و bootstrap.css یافت می شوند. اینها تصاویر پس زمینه قابل استفاده مجدد هستند که ما برای کارت ها استفاده می کنیم. می توانید این تصاویر را با جستجوی آنها در CSS و جایگزینی آنها با تصاویر خود استفاده کنید.

Bg-1 تا Bg-2

کلاس های کمکی اضافی

ما همچنین برخی از کلاس های کمکی اضافی داریم که می توانید از آنها استفاده کنید، که واقعاً نمی توانیم آنها را به هیچ یک از عناصر بوت استرپ اضافه کنیم. این ابزارها یا کلاس هستند یا در صورتی که **data-** را در جلوی آنها می بینید، **attributes** هستند.

data-toggle-theme-switch – تغییر تم قالب از روشن به تاریک و بالعکس

border-0 – حاشیه را از یک عنصر حذف می کند

لطفاً فراموش نکنید که کلاس های Bootstrap را نیز بررسی کنید. کلاس های ثانویه زیادی وجود دارد که می توانید در کنار کلاس ما از آنها استفاده کنید. ما به تازگی چارچوب بوت استرپ را گسترش داده ایم تا همه کاربردی ترین برنامه های افزودنی را شامل شود.

getbootstrap.com/docs/5.3/utilities/spacing

نتیجه گیری

ایده ایجاد چیزی انعطاف پذیر همیشه در ذهن ما بود و از بازخورد مشتریان ما به راحتی می توان فهمید که اکثر کاربرانی که فقط می خواهند یک راه اندازی سریع برای PWA خود داشته باشند، نمی دانند و نمی خواهند از CSS استفاده کنند. این به کاربران این امکان را می دهد که یک طراحی کپی پیست با حداقل استفاده از CSS داشته باشند. همه چیز را می توان با استفاده از کلاس ها در حین نوشتن HTML ایجاد کرد و همه چیز در محصول ما را می توان با این عناصر بهبود بخشید.

هایلایت رنگ و پس زمینه ها

قدرت و سهولت رنگ ها

هنگام ایجاد رنگ برای قالب، ما همیشه به انعطاف پذیری فکر می کنیم و به شما این امکان را می دهیم که به راحتی هر طرح رنگی را که می خواهید بدون مشکل ایجاد کنید. بنابراین، استفاده از **SCSS** بهترین راه حل برای اطمینان از یک طرح رنگی سریع، آسان و قابل اعتماد و در عین حال ایجاد طرح های رنگی جدید فوق العاده آسان بود.

۳ مرحله ساده برای ایجاد یک طرح رنگی جدید با استفاده از **SCSS** و یک مرحله برای اعمال هایلایت اصلی در کل صفحه وجود دارد.

چگونه می توانم **color-highlight** و **bg-highlight** را در سطح قالب تغییر دهم؟

مثل همیشه ساده است. به سادگی به بالای فایل html خود بروید و "body data-highlight=highlight-COLOR" را به رنگ مورد نظر خود تغییر دهید. لطفاً مرحله زیر را بخوانید تا متوجه شوید که قالب چگونه طرح رنگ صحیح را درک و بارگذاری می کنند. ما یک راهنمای کوچک از طریق این لینک برای کمک به شما در درک نحوه ایجاد رنگ های SCSS و فایل های SCSS آورده ایم. استفاده از آن واقعاً ساده است!

اکنون می توانید از کدهای تولید شده بالا در موارد زیر استفاده کنید

color-awesomeBuyer-light – به متن و آیکون ها اعمال می شود، از hex1 استفاده می کند

color-awesomeBuyer-dark – به متن و آیکون ها اعمال می شود، از hex2 استفاده می کند

bg-awesomeBuyer-light – به عناصر پس زمینه مانند کارت اعمال می شود، از hex1 استفاده می کند

bg-awesomeBuyer-dark – برای عناصر پس زمینه مانند کارت اعمال می شود، از hex2 استفاده می کند

bg-fade-awesomeBuyer-light – نسخه محو شده رنگ **bg-awesomeBuyer-light** بالا

bg-fade-awesomeBuyer-dark – نسخه محو شده رنگ **bg-awesomeBuyer-dark** بالا

gradient-awesomeBuyer – برای عناصر پس زمینه مانند کارت کاربرد دارد، از ترکیبی از هر ۳ hex استفاده می کند که با hex1، hex2، hex3 در پایان شروع می شود.

border-awesomeBuyer – برای عناصر حاشیه دار با استفاده از hex1 اعمال می شود

border-fade-awesomeBuyer – نسخه محو شده رنگ **border-awesomeBuyer** بالا

چه رنگ هایی به طور پیش فرض گنجانده شده است؟

اگر می خواهید از این رنگ ها در تگ body یا تگ هایلایت استفاده کنید، فقط باید نام آنها را مطابق شکل زیر وارد کنید، اما اگر می خواهید از این رنگ ها برای تغییر رنگ یک عنصر یا پس زمینه یک عنصر استفاده کنید. از ساختار awesomeBuyer که در بالا توضیح داده شد استفاده کنید و به سادگی awesomeBuyer را با هر رنگی از لیست زیر جایگزین کنید.

نکته:

همه رنگ‌های قالب را می‌توان در داخل فایل `scss/appkit/_colors.scss` یا داخل `style.css` با جستجو بر اساس فرمول زیر پیدا کرد.

bg-ANY_NAME_FROM_LIST_BELOW-dark

orange •	green •	mint •
pumpkin •	sky •	plum •
pink •	blue •	violet •
Brown •	yellow •	Magenta •
dark •	teal •	red •

color-theme و bg-theme چیست؟

فرض کنید یک کارت سیاه دارید که در آن یک متن سفید نوشته شده است. وقتی حالت تاریک را فعال می‌کنید، آن متن سیاه می‌شود و دیگر قابل خواندن نیست. اینجا است که کلاس **color-theme** عالی است. **color-theme** و **bg-theme** همیشه رنگ مخالف تمی هستند که انتخاب کرده‌اید.

color-theme – theme-dark تاریک و **bg-theme** سفید است

color-theme – theme-light روشن و **bg-theme** سیاه است

منوها

مودال ها، نوارهای کناری و پاپ آپ

سیستم منو به‌طور سفارشی ساخته شده‌اند تا دارای افکت‌ها و رابط‌های زیباتر از سیستم‌های پیش‌فرض موجود در بوت استرپ باشند. آن‌ها برای استفاده کامل از CSS بسیار خوب بهینه‌سازی شده‌اند و حتی بر روی موبایل‌های قدیمی نیز بدون نقص کار خواهند کرد.

منوهای نشان داده شده در اینجا سفارشی ساخته شده‌اند. مدال‌هایی که بوت استرپ ارائه می‌دهد با صفحات قالب کاملاً سازگار هستند اما نمی‌توانند با منوهای ما ترکیب شوند. همانطور که در اینجا با استفاده از سیستم آنها نشان داده شده است، همچنان می‌توانید از مدال‌های بوت استرپ استفاده کنید:

getbootstrap.com/docs/5.3/components/modal

منوها باید خارج از **div class="page-content"** قرار گیرند تا اطمینان حاصل شود که می توانند به خوبی محتوا را مرور کنند و در صورت استفاده از افکت ها می توانند محتوای صفحه را بدون شکستن آن کنترل کنند. لیست موارد زیر باید خارج از کلاس **page-content** قرار گیرد

- هر آیتمی با کلاس "menu..."
- action sheets
- action modals
- sidebars
- snackbars
- toasts

ساختار کلی و عمومی منوها

نمونه های جدا پس از این بخش گنجانده شده است

اول از همه، بیایید یک ساختار ساده، بدون هیچ گونه تنظیمات اضافی را در نظر بگیریم. برای ساده تر خواندن و دنبال کردن، هر ویژگی را در مثال زیر در یک خط جداگانه ساختیم تا بتوانیم آنها را با شماره توضیح دهیم.

```
1. <div id="MENU_UNIQUE_NAME"
2. class="menu MENU_STYLES MENU_BOX_POSITION"
3. data-menu-effect="MENU_EFFECT">
4. <!-- Place your menu, action sheet, action modal stuff here -->
5. </div>
```

خط ۱ شناسه منو است. این باید منحصر به فرد باشد تا بتوانیم از آن برای فراخوانی آن از دکمه HTML مورد نظر خود یا مستقیماً از جاوا اسکریپت استفاده کنیم.

خط ۲ جایی است که ما نوع منویی را که می خواهیم تعریف می کنیم و همچنین سبکی به آن اضافه می کنیم. ابتدا منوی کلاس را اضافه کنید که ضروری است، اکنون می توانید جایگزین کنید.

- **MENU_STYLES** - کلاس های کاربردی مانند rounded-VALUE یا shadow-VALUE برای تغییر شکل منو. حتی می توانید با افزودن کلاس **menu-box-detached**، کاری کنید که منو از کناره های صفحه جدا شود، یعنی حس شناور بودن را داشته باشد.
- **MENU_BOX_POSITION** - برای کنترل محل قرارگیری منو
- **menu-box-left** - برای ایجاد یک نوار کناری سمت چپ

- **menu-box-right** - برای ایجاد یک نوار کناری سمت راست
- **menu-box-bottom** - برای ایجاد منوی پایین
- **menu-box-top** - برای ایجاد یک منوی بالا
- **menu-box-modal** - برای ایجاد یک پنجره مودال

خط ۳ فقط در صورتی استفاده می شود که از `menu-box-top`, `menu-box-bottom`, `menu-box-modal` استفاده می کنید. مقدار را با مقداری که می خواهید منوی شما بالا باشد جایگزین کنید.

خط ۴ فقط در صورتی استفاده می شود که از `menu-box-left`, `menu-box-right` استفاده می کنید. مقدار را با مقداری که می خواهید منوی شما گسترده باشد جایگزین کنید.

خط ۵ افکتی است که می خواهید منوی شما داشته باشد. این برای `menu-box-modal` صدق نمی کند

- **MENU_EFFECT** - از هر یک از جلوه های منوی زیر پشتیبانی می کند:
- **menu-push** - محتوای صفحه را با ارتفاع یا عرض منوی انتخابی شما فشار می دهد
- **menu-parallax** - محتوای صفحه را به اندازه ۱۰/۱ اندازه منو فشار می دهد و افکت پارالاکس ایجاد میکند

خط ۶ اینجا جایی است که تمام محتوایی را که می خواهید نشان دهید در منوی خود قرار می دهید.

خط ۷ جایی است که کل منو بسته میشود

حالا ببایید چند مثال نشان دهیم تا همه متوجه شوند

چگونه می توانم منویی مانند این را فعال یا نشان دهم؟

برای فعال کردن منو، تنها کاری که باید انجام دهید این است که ویژگی `"data-menu="MENU_UNIQUE_NAME"` را به دکمه مورد نظر خود اضافه کنید. هر منو باید یک نام منحصر به فرد در خط ۱ داشته باشد و آن نام به منوی داده در دکمه ای که می خواهید منو از آن فعال کنید اضافه می شود.

چگونه می توانم یک منو را به صورت کدنویسی نشان دهم یا پنهان کنم؟

در `custom.js`، در داخل تابع `init_template()` می توانید منوها را با استفاده از کدهای زیر فعال کنید.

کد باید تابع منو را داشته باشد و به دنبال آن شناسه منحصر به فرد منو، تابع، نمایش یا پنهان کردن و زمان پایان قبل از نمایش منو، به صورت میلی ثانیه بیان شده باشد.

```
1. //show menu
2. menu('your-menu-id', 'show', 250);
3. //hide menu
4. menu('your-menu-id', 'hide', 250);
```

آیا می توانم منوها را از طریق منوها فعال کنم؟

بله، در منوی خود، دکمه ای با `data-menu="your-menu-id"` اضافه کنید و آن منو فعال می شود.

چگونه یک منو را از داخل منو ببندم؟

کلاس `close-menu` را به هر دکمه ای در منوی خود اضافه کنید. این باعث بسته شدن آن می شود.

مودال و منوهای بوت استرپ

آیا هنوز هم می توانم از منوهای پیش فرض بوت استرپ استفاده کنم؟

همچنان می توانید از سیستم منوی بوت استرپ استفاده کنید، آن ها می توانند به صورت برنامه نویسی مانند آنچه در بوت استرپ بیان شده است راه اندازی و فعال شوند، اما لطفاً موارد زیر را در نظر داشته باشید.

⚠ همه کدهای HTML منو را خارج از کلاس محتوای صفحه قرار دهید
⚠ کد فعال کننده خود را در داخل `custom.js` داخل تابع `init_template()` قرار دهید

منو اصلی

استفاده از منوی اصلی در قالب

لطفا توجه داشته باشید که منوهای اکسترنال از طریق جاوا اسکریپت بارگیری می شوند. این بدان معنی است که آنها برای مشاهده به صورت لوکال به یک محیط سرور، مانند AMPPS، XAMP یا WAMP نیاز دارند. اگر محیط سرور ندارید، لطفاً از code.visualstudio.com استفاده کنید. این یک ویرایشگر کد ۱۰۰٪ رایگان است که دارای گزینه "پیش نمایش زنده" است که سرورهای لوکال را شبیه سازی می کند. اگر منو سفید بارگذاری شده است، دلیل این امر است. مرحله بالا آن را برطرف می کند. این در سرور شما یا هنگام استفاده از صفحه به عنوان PWA اتفاق نمی افتد.

با توجه به اینکه اپکیت یک قالب HTML است، ویرایش کل منو به معنای ویرایش هر صفحه HTML جداگانه است که برای صفحات بزرگ می تواند بسیار پیچیده باشد. به همین دلیل، قالب از منوهای خارجی (external) استفاده میکند که ویرایش آن را بسیار ساده تر می کند. شما فقط باید یک فایل HTML را ویرایش کنید. بیایید توضیح دهیم:

```
1. <div
2.   id="menu-main"
3.   class="menu menu-box-left"
4.   data-menu-width="280"
5.   data-menu-active="nav-welcome"
6.   data-menu-load="menu-main.html">
```

```
7.  
8. <!-- (1) -->  
9.  
10. </div>
```

کد بالا یک تگ div است، ما آن را به چند خط تقسیم می‌کنیم تا مشاهده و خواندن آن آسان‌تر شود.

خط ۲ id منو است. این توسط دکمه "data-menu="menu-main" هدف قرار خواهد گرفت.

خط ۳ و ۴ - نوع منو و تنظیمات مربوط به آن است.

خط ۵ خط ۵ کلاس فعالی است که می‌خواهیم استفاده کنیم. در داخل menu-main.html خواهید دید که هر لینک

دارای یک id منحصر به فرد است. در مثال بالا، id=nav-welcome را فعال می‌کند. می‌توانید از این روش در

صفحات خود برای "انتخاب" یا "فعال کردن" صفحه‌ای که مشاهده می‌کنید استفاده کنید. لطفاً توجه داشته باشید

که این فقط برای menu-main.html اعمال می‌شود و فقط برای **یک** منو می‌توان از آن استفاده کرد.

خط ۶ منویی است که می‌خواهید بارگیری کنید. در این مورد ما می‌خواهیم منوی menu-main.html را بارگیری کنیم.

این بدان معنی است که پس از بارگیری صفحه، محتوای menu-main.html در این div بارگیری می‌شود.

خط ۸ - اگر نمی‌خواهید از منوهای خارجی استفاده کنید، کافی است محتویات menu-main.html یا منویی را که در

حال بارگذاری در <!-- (1) --> هستید کپی و پیست کنید و خط ۶ را حذف کنید. به این صورت است که از منوی

همان صفحه HTML استفاده می‌کنید و آن را از منبع خارجی بارگیری نمی‌کنید.

PWA

استفاده از قالب به عنوان یک وب اپلیکیشن

یک وب اپلیکیشن اساساً به این معنی است که می‌توانید وب سایت را در صفحه اصلی خود "نصب" کنید، یک آیکون

داشته باشید و آن را دقیقاً شبیه به یک برنامه بومی نشان دهید. با این حال، این موارد با چیزهایی همراه است که

شما باید بدانید.

الزامات

برای اینکه وب سایت شما به عنوان PWA فعال شود، به این معنی که درخواست نصب آن را در صفحه اصلی خود

نشان دهید، صفحه شما باید معیارهای زیر را داشته باشد.

- حداقل Android ۷ (ترجیحاً ۸) یا iOS ۱۲ را اجرا کنید
- سایت شما باید دارای گواهی SSL باشد
- فایل _manifest.json و _service-work.js را در همان پوشه فایل های HTML قرار دهید.

وب اپلیکیشن برای اندروید و iOS

دستگاه‌های اندرویدی یکپارچه‌سازی بسیار بهتری برای PWA دارند تا نگهداری و استفاده از آنها آسان‌تر شود. در مورد اندروید، هر زمان که وب اپلیکیشن را باز می‌کنید، محتوا به طور خودکار به روز می‌شود و همیشه محتوای تازه را در اختیار شما قرار می‌دهد.

در مورد iOS، اپل با قدرت تمام به وب اپلیکیشن‌ها قدرت واقعی را که شایسته آن هستند، می‌دهد. اپل تنها با iOS ۱۲ شروع به خواندن فایل‌های manifest و پذیرش سرویس‌دهندگان کرد، و حتی پس از آن، با مشکلات کوچک. در مورد iOS، دستگاه یک کپی را روی خود کش می‌کند.

قالب به عنوان وب اپلیکیشن نصب نمیشود؟

در صورتی که نمی‌توانید جعبه‌های PWA را ببینید، اگرچه همه چیز در سرور شما کپی شده است، ۲ ویرایش وجود دارد که باید انجام شود. این ممکن است برای برخی از هاست‌ها بدون دلیل اتفاق بیفتد، یا ممکن است برای میزبان‌های دیگر اصلاً مورد نیاز نباشد.

در manifest.json محدوده خود را تنظیم کنید و لینک را به مقصد فایل‌های خود شروع کنید: آدرس‌های سرور ما را با همان ساختاری که در زیر ارائه می‌شود جایگزین آدرس‌های خود کنید.

```
1. "scope" : "https://www.your-domain.com/yourfolder/",
2. "display" : "fullscreen",
3. "start_url" : "https://www.your-domain.com/yourfolder/index.html",
```

در custom.js، در بالای صفحه، کد زیر را خواهید دید.

```
1. var pwaScope = "/";
2. var pwaLocation = "/_service-worker.js";
```

همانطور که در زیر نشان داده شده است، کد بالا را با آدرس کامل صفحه و سرویسکار خود جایگزین کنید.

```
1. var pwaScope = "https://your-domain.com/your-folder/";
2. var pwaLocation = "https://your-domain.com/your-folder/_service-worker.js";
```

افزودن پشتیبانی آفلاین

لطفاً توجه داشته باشید که پشتیبانی آفلاین بر روی جدیدترین سیستم عامل‌ها بهترین عملکرد را دارد. برخی از سیستم‌عامل‌های قدیمی‌تر اندروید و iOS می‌توانند مشکلاتی در این مورد ایجاد کنند. برای اینکه برنامه خود را به صورت آفلاین اجرا کنید، باید تمام فایل‌های خود را کش کنید.

برای کش کردن تمام دارایی‌ها، service-worker.js را باز کنید و در جایی که آرایه REQUIRED_FILES را می‌بینید، تمام تصاویر، اسکریپت‌ها، استایل و فایل‌های HTML را که برنامه‌تان استفاده می‌کند اضافه کنید.

لطفاً توجه داشته باشید

نحوه وارد کردن فایل‌ها را بررسی کنید و نمونه‌های دقیق را دنبال کنید. اگر هر گونه خطای متنی (اشتباهات تایپی) دارید یا مورد را به درستی لینک‌ها را وارد نمی‌کنید، قالب برای استفاده آفلاین فایل‌ها را کش نمی‌کند! همچنین

مطمئن شوید که مراحل بالا را انجام داده اید و `pwaScope` و `pwaLocation` شما در `custom.js` تنظیم شده اند و همچنین `scope` و `start_url` در `manifest.json` شما نیز تنظیم شده است. بدون تنظیم صحیح این موارد، PWA شما فایل‌ها را به صورت آفلاین ذخیره نمی‌کند.

چگونه می‌توانم کش PWA خود را پاک کنم؟

برای پاک کردن کش هنگام ایجاد تغییرات، کافی است `service-worker.js` را باز کنید و شماره `APP_VER` را تغییر دهید. با هر تغییر آن را افزایش دهید. پس از انجام این کار و بسته شدن و باز کردن مجدد برنامه، محتوای جدید را نشان می‌دهد. گاهی اوقات، ۲ یا ۳ رفرش مورد نیاز است.

برای اینکه کش خود را همیشه پاک نگه دارید، به سادگی به `custom.js` بروید و `pwaNoCache = true` را در بالای `custom.js` تنظیم کنید. در حال حاضر این روی `false` تنظیم شده است. توصیه می‌کنیم هنگام توسعه، این مقدار را `true` نگه دارید و سپس آن را به `false` برگردانید. بهتر است فقط `APP_VER` را به‌روزرسانی کنید تا در مواقع ضروری، دستگاه‌ها را مجبور نکنید داده‌ها را بدون دلیل دانلود کنند.

المان‌ها و صفحات

صدها المان و صفحات انعطاف پذیر در انتظار شماست.

در شخصی سازی قالب گیر کردی؟ جای نگرانی نیست!

اگر مشکلی دارید که در اینجا ذکر نشده است، لطفاً از طریق تیکت در راست چین در ارتباط باشید. بنده همیشه پاسخگو هستم، در کمتر از ۳ ساعت به همه سوالات شما پاسخ می‌دهم.

ارسال تیکت