

بِسْمِ خُدا



دانشگاه پیام نور استان تهران

مرکز تهران / واحد ری

پروژه کارشناسی

رشته مهندسی کامپیوتر

گرایش نرم افزار

عنوان پروژه:

طراحی و اجرای وبسایت واکنشگرایی کاربردی

استاد راهنما

جناب آقای دکتر سید علی ابراهیمی رضوی

تهیه کننده:

محمد رضا ناطقی

زمستان ۱۴۰۰

قدردانی و سپاس

پروردگار متعال را شاکر و سپاسگزارم که توانستم این مرحله از تحصیل را با راهنمایی های اساتید بزرگوار به پایان برسانم.

بر خود لازم می دانم که از استاد راهنمای ارجمند جناب آقای **دکتر سید علی ابراهیمی رضوی**

که با راهنمایی های دقیق و ارزنده خویش در طول تحصیل و اجرای این پروژه همواره بنده را یاری نموده و حق استادی را بر اینجانب تمام کردند سپاسگزاری کنم.

چکیده:

طراحی وب سایت امروزه جایگاه ویژه ای در میان عناوین نوین مرتبط با تکنولوژی و شغلی دارد، بطوری که حتی بهترین تکنولوژی ها و فن آوری های روز بدون تمرکز برای ایجاد رابطه کاربری و تجربه کاربری مناسب نمی توانند جایگاه خود را در میان کاربران (که شرط اصلی برای موفقیت در انواع پروژه های تجاری است) را به دست بیاورند. دلیل اصلی برای طراحی وب سایت ساده تر کردن دسترسی اطلاعات برای کاربران و آسان تر شدن برقراری ارتباط بین آنها و تجربه آنها در تعامل با وب سایت است و در واقع بخشی پیشرو و همه گیر شده ای از روابط انسان و کامپیوتر در دنیای امروز به وسیله طراحی وب در جریان است.

طراحی واکنش گرا در واقع با توجه به سخت افزار و ابعاد صفحه نمایش کاربر به تغییر بخش های مختلف وب سایت برای ایجاد تعامل هرچه بیشتر کاربر و در نتیجه آسان تر شدن هرچه بیشتر رابطه کاربر با وب سایت و تجربه او از استفاده از صفحه مورد نظر ایجاد شده است.

در این پروژه سعی شده تا چرایی، تاریخچه، قوانین و ابزارهای طراحی وب سایت و همچنین نحوه اجرای طرح بوجود آمده با توجه به استانداردهای امروزی که در این حیطه مورد استفاده قرار می گیرد به ساده ترین شکل بدون حذف نکات اصلی در اختیار خواننده قرار بگیرد.

فهرست مطالب

صفحه	عنوان
ث	قدردانی و سپاس
ج	چکیده
ج	فهرست مطالب
	فصل اول: مفاهیم و اهداف
۱	۱-۰ - طراحی وب و ارتباط آن با تعامل انسان و کامپیوتر
۱	۱-۱ - طراحی وب چیست؟
۲	۱-۲ - طراحی وب، زبان های مخصوص به خود را دارد
۳	۱-۳ - طراحی وب
۴	۱-۴ - مراحل کلی طراحی وب سایت چیست؟
۵	۱-۵ - انواع طراحی سایت
۶	۱-۶ - اهداف تحقیق
	فصل دوم: تاریخچه طراحی وب
۷	۲-۱ - تعامل انسان و کامپیوتر
۷	۲-۲ - تعامل انسان و کامپیوتر و رابط کاربری
۸	۲-۳ - اصول طراحی رابط کاربری
۹	۲-۴ - تاریخچه طراحی وب سایت
	فصل سوم: اصول و چهارچوب های طراحی وب
۱۴	۳-۱ - اصول طراحی
۱۹	۳-۲ - چیدمان وبسایت
۲۱	۳-۳ - مسیریابی (نویگیشن)
۲۳	۳-۴ - UX
۲۳	۳-۵ - طراحی ادپتیو در مقابل طراحی ریسپانسیو

فصل چهارم: اجرا

۲۵	۱_۴ - طرح اولیه و جمع آوری داده ها
۲۷	۲_۴ - کد ابتدایی HTML
۳۰	۳_۴ - کد ابتدایی CSS
۳۱	۴_۴ - کد ابتدایی JavaScript
۳۲	۵_۴ - اضافه کردن آیکون ها، فونت ها و رنگ بندی
۳۵	۶_۴ - هدر و منوی نوگیشن
۴۰	۷_۴ - طراحی کانتینر HOME
۴۳	۸_۴ - کانتینر Social
۴۵	۹_۴ - کانتینر Profile
۴۵	۱۰_۴ - کانتینر Education
۴۷	۱۱_۴ - کانتینر Skills
۴۹	۱۲_۴ - کانتینر Experience
۵۰	۱۳_۴ - کانتینر Certificates
۵۱	۱۴_۴ - کانتینر References
۵۲	۱۵_۴ - کانتینر Interests
۵۳	۱۶_۴ - کد منو و انیمیشن Scroll top
۵۴	۱۷_۴ - بخش Media Queries - تکمیل واکنشگرایی سایت
۵۶	۱۸_۴ - مود های تصویر روز و شب (button Dark/Light)
۵۸	۱۹_۴ - نکته نهایی

۵۹ فصل پنجم: نتیجه گیری و نکات نهایی

۶۲ منابع

فصل اول: مفاهیم و اهداف

۱-۰ - طراحی وب و ارتباط آن با تعامل انسان و کامپیوتر

بطور کلی طراحی وب حیطه ای از رشته ای جدید به نام تعامل انسان و کامپیوتر "HCI" در نظر گرفته می شود.

تعامل انسان و کامپیوتر یک حوزه مطالعاتی بین رشته ای است که بر روی طراحی تکنولوژی کامپیوتر و به ویژه تعامل بین انسان ها یا کاربران با کامپیوترها تمرکز دارد. اگر چه این حوزه در ابتدا فقط با کامپیوترها سر و کار داشت، مطالعات آن اما به تدریج شامل طراحی تکنولوژی اطلاعاتی در تقریبا همه شکل های آن شده است؛ به عنوان یک حوزه مطالعاتی بین رشته ای، این حوزه با رشته های علوم کامپیوتر، علوم رفتاری، طراحی، مطالعات رسانه و چندین حوزه علمی دیگر پیوند دارد.

۱-۱ - طراحی وب چیست؟

ایجاد ظاهر وبسایت (نرم افزار تحت وب) با بهره گیری از طراحی گرافیک که معمولا پیاده سازی آن در نرم افزارهای طراحی انجام می پذیرد؛ و به وسیله زبان های طراحی تحت وب یعنی HTML, CSS به فرمتی برای اجرا در مرورگر تبدیل می شود.

در تعریف بالا چند نکته مشخص شد:

۱. طراحی وب، ظاهر یک نرم افزار تحت وب است.

۲. طراحی وب، زبان های مخصوص به خود را دارد.

۳. طراحی وب، دو مرحله اصلی دارد.

۱- طراحی وب، ظاهر نرم افزار تحت وب است

آنچه بصورت بصری (دیداری) در یک نرم افزار تحت وب قابل مشاهده است ظاهر آن تلقی می شود. از همین رو طراحی وب، عملی برای ایجاد پوسته ظاهری نرم افزار تحت وب می باشد.

۱-۲- طراحی وب، زبان های مخصوص به خود را دارد

برای یادگیری طراحی تحت وب باید زبان های مخصوص به آن را فرا گرفت. زیرا در محیط کدنویسی ایجاد می شوند. این زبان ها JavaScript, CSS, HTML می باشند.

۱-۲-۱- اچ تی ام ال چیست؟

Hyper Text Markup Language که به اختصار HTML نامیده می شود؛ به معنی زبان نشانه گذاری ابر متن است. یک فایل متنی ساده است که به این زبان نوشته (کدنویسی) می شود و با پسوند html. ذخیره شده و برای مرورگر قابل مشاهده است HTML. ساختار و بدنه اصلی طراحی تحت وب به حساب می آید. این زبان بر اساس وجود تگ هایی از پیش تعیین شده، نوشته شده و هر یک از تگ ها معرف عملکردی خاص است.

۱-۲-۲- سی اس اس چیست؟

Cascading Style Sheet که به اختصار CSS نامیده می شود؛ به معنی برگه سبک آبشاری است. زبانی است که بصورت مستقل برای سبک دهی و ایجاد جلوه های مختلف بر روی عناصر موجود در HTML مورد استفاده قرار می گیرد. در واقع CSS کار زیباسازی را در طراحی وب بر عهده دارد.

۱-۲-۳- جاوا اسکریپت چیست؟

JavaScript که به اختصار JS نامیده می شود؛ زبانی شبه برنامه نویسی است که پویاسازی یک وبسایت را با استفاده از تغییر در عناصر بر عهده می گیرد. به این دلیل آن را یک زبان شبه برنامه نویسی می نامند چون این زبان

در سمت کاربر و در مرورگر قابلیت اجرا داشته و نیازی به پایگاه داده برای ذخیره سازی، یا یک سرور برای اجرا ندارد.

البته با گسترش موتورهای این زبان برنامه نویسی، این زبان آرام آرام در حال حذف رقبا است و می تواند با استفاده از فریم ورک هایی مثل مانگو دی بی و یا نود جی اس به زبانی با عملکرد بالا در تمامی لایه های وب سایت تبدیل شود و مورد استفاده قرار بگیرد.

۱-۳- طراحی وب

۱-۳-۱ - طراح وب کیست؟

فردی که دانش و تجربه استفاده از زبان های طراحی وب و توان تکمیل یک پروژه عملی را داشته باشد طراح وب نامیده می شود.

۱-۳-۲ - وظایف طراح وب چیست؟

وظایف یک طراح سایت به این بستگی دارد که مسئولیت طراحی و ساخت کدام قسمت سایت را بر عهده دارد. برای راه اندازی یک سایت حرفه ای، به دو تیم یا دو شخص نیاز است. یک تیم برای طراحی شکل و ظاهر سایت و یک تیم برای طراحی کدهای سرور و کدهایی که برای کاربران قابل دیدن نیستند.



۱-۴ - مراحل کلی طراحی وب سایت چیست؟

مرحله ۱: طراحی وایر فریم

ابتدا با توجه به نوع سایت و نیاز مشتری، یک نمای کلی از سایت طراحی می‌کنیم. این نما شامل یک نقشه کلی از صفحات سایت و تعیین جایگاه و چیدمان المان‌ها و عناصر مختلف است. این مرحله از طراحی وب سایت «طراحی وایر فریم» نام دارد.

مرحله ۲: طراحی گرافیک سایت

پس از مشخص شدن محل قرارگیری عناصر سایت، باید سایت را طراحی کنیم. در این مرحله با توجه به نوع وب سایت، با استفاده از نرم‌افزارهای مخصوص رنگ‌بندی و گرافیک سایت طراحی می‌شود.

مرحله ۳: کدنویسی و برنامه‌نویسی (تمرکز اصلی تحقیق با توجه به رشته)

با توجه به سایت طراحی شده و نیاز مشتری، در این مرحله می‌توان از روش‌های مختلفی برای مدیریت سایت استفاده کرد. سیستم‌های مدیریت محتوا مانند وردپرس و یا طراحی یک فریم ورک اختصاصی برنامه‌نویسی. تصمیم‌گیری در این بخش براساس هزینه‌ها و نیاز مشتری صورت می‌گیرد.

مرحله ۴: راه‌اندازی سایت

پس از انجام ۳ مرحله قبل، به راه‌اندازی سایت در بستر جهانی اینترنت می‌پردازیم. در این مرحله پس از تهیه یک «هاست» و «نام دامنه مناسب»، سایت را پیکربندی کرده و در دسترس عموم قرار می‌دهیم.

۱-۵ - انواع طراحی سایت

در قسمت قبل با مفهوم Frontend (فرانت‌اند) و Backend (بک‌اند) سایت آشنا شدیم. اکنون به بررسی انواع طراحی وب سایت می‌پردازیم.

سایت‌ها به دو دسته کلی ایستاتیک (Static Website) و دینامیک (Dynamic Website) تقسیم می‌شوند.

۱_۵_۱ - سایت‌های ایستاتیک یا ایستا:

تنها قسمت فرانت‌اند انجام می‌شود، یعنی مطالب و محتویات سایت تنها یک بار جمع‌آوری شده و هرکدام از صفحات سایت، جداگانه طراحی و بارگذاری می‌شوند. در این سایت‌ها هرگونه تغییر باید توسط طراح و کارشناس سایت انجام شود. از این مدل معمولاً برای طراحی وب سایت‌های معرفی (که تا مدت‌ها محتوای آن تغییر نمی‌کند) استفاده می‌شود.

۲_۵_۱ - سایت دینامیک یا پویا:

در طراحی این سایت‌ها هر دو بخش فرانت‌اند و بک‌اند برنامه‌نویسی می‌شوند. سایت‌های پویا نیاز به تغییر مداوم محتوا دارند و با توجه به کاربرد سایت، طراحی شده و توسعه پیدا می‌کنند. در سایت‌های دینامیک برنامه‌نویس بک‌اند، وظیفه پویاسازی وب‌سایت را بر عهده دارد و باید اسکریپتی را طراحی کند که سایت به‌طور کامل پویا باشد.

در دنیای امروز تقریباً برای هر هدف، یک سایت طراحی می‌شود تا به‌عنوان پایگاه اطلاع‌رسانی، فروش و... مورد استفاده قرار گیرد. موفق‌ترین این سایت‌ها آن‌هایی هستند که اصول طراحی و بهینه‌سازی سایت را به‌درستی رعایت کرده‌اند تا محتوا را به مخاطبان خود نمایش دهند. طراحی و ساخت یک وب‌سایت کارآمد و موفق، این امکان را فراهم می‌کند که در کمترین زمان، جدیدترین محصولات به‌همراه مشخصات کامل آن، در اختیار مخاطبان قرار گیرد، این موضوع، رشد مشتریان و افزایش سود فعالیت اقتصادی را به‌دنبال خواهد داشت.

۱_۶ - اهداف تحقیق

بطور کلی هدف ما از انجام این تحقیق آشنایی با متدها و ابزارهای روز برای ایجاد سایتی بسیار ساده و استاتیک است که خود اولین قدم برای شروع طراحی وب سایت به صورت حرفه ای است. مهم ترین نکته رعایت شده واکنشگرا بودن این وب سایت است که امروزه کاربرد بسیار زیادی دارد و از مهارت های مهم در طراحی هر نوع وب سایتی به حساب می آید همچنین این وب سایت می تواند برای اطلاعاتی همچون رزومه و یا معرفی شرکت و یا محصولی خاص مورد استفاده قرار بگیرد با این حال تمرکز اصلی بر روی رزومه فردی است.

فصل دوم: تاریخچه طراحی وب

۱-۲ - تعامل انسان و کامپیوتر

اصطلاح تعامل انسان و کامپیوتر برای اولین بار توسط استوارت کی کارد، آلن نول، و تامس پی موران در کتاب «روانشناسی تعامل بین انسان-کامپیوتر» که در سال ۱۹۸۳ چاپ شد محبوبیت یافت. این اصطلاح به این موضوع اشاره دارد که برخلاف ابزارهای دیگر که کاربردهای محدودی دارند، یک کامپیوتر کاربردهای گسترده‌ای دارد و این مانند یک مکالمه با پایان باز بین کاربر و کامپیوتر اتفاق می‌افتد. این تصور مکالمه باعث تشبیه تعامل بین انسان-کامپیوتر با تعامل بین انسان‌ها می‌شود، مقایسه‌ای که برای تحلیل‌های نظری در این حوزه از اهمیتی کلیدی برخوردار است.

۲-۲ - تعامل انسان و کامپیوتر و رابط کاربری

رابط کاربری بین انسان و کامپیوتر اغلب به عنوان نقطه ارتباط بین کاربر انسانی و کامپیوتر شناخته می‌شود. گردش اطلاعات بین این دو اغلب تحت عنوان حلقه‌ی تعامل (Loop of Interaction) شناخته می‌شود؛ این حلقه یک بعدی نیست و شامل چندین جنبه مجزا می‌شود، بعضی از این جنبه‌ها شامل موارد زیر می‌شود:

- **دیداری:** تعامل دیداری انسان و کامپیوتر.
- **شنیداری:** تعامل شنیداری انسان و کامپیوتر.
- **محیط وظایف:** شرایط و اهدافی که از طرف کاربر تعیین می‌شود.
- **محیط ماشین:** محیطی که کامپیوتر در آن قرار دارد.

- **محدوده‌های رابط کاربری:** فرایندهایی که به یک تعامل اختصاص دارند.
- **گردش ورودی:** گردش اطلاعاتی که هنگام انجام وظایف شروع می‌شود.
- **خروجی:** گردش اطلاعاتی که از محیط ماشین آغاز می‌شود.
- **بازخورد:** حلقه‌هایی از رابط کاربری، که فرایندها را ارزیابی، تحلیل و تأیید می‌کنند.
- **هماهنگی:** مطابقت بین طراحی کامپیوتری، کاربر و وظیفه‌ای که به کامپیوتر منتقل می‌شود.

مطالعات تعامل انسان و کامپیوتر در هر یک از زمینه‌های بالا به دنبال درک نحوه ارتباط انسان با ماشین‌های کامپیوتری و بهبود سیستم‌های این حوزه است؛ به نقل از ویکی‌پدیا، تعامل انسان و رایانه به دانش و فناوری مدرن و پرتنوع مطالعه، طراحی، اجراء، و ارزیابی سامانه‌های محاسباتی درگیر در محاورات و تعاملات مابین کاربران انسانی از یک سو، و رایانه‌ها و عامل‌های هوشمند نرم‌افزاری از سوی دیگر می‌گویند؛ این رشته، علم بررسی تعامل انسان و رایانه است؛ در واقع این علم نقطه تقاطع علوم رایانه و علوم رفتارشناسی طراحی و چند علم دیگر است، ارتباط و تعامل انسان و رایانه از طریق واسطه اتفاق می‌افتد؛ که شامل نرم‌افزار و سخت‌افزار است.

۲-۳ - اصول طراحی رابط کاربری

از این استانداردها در هنگام طراحی رابط کاربری (User Interface) کلاینت استفاده می‌شود "مقاومت، سادگی، قابل درک بودن، قابل تأمین بودن، ثبات، ساختار و بازخورد از جمله‌ی این استانداردهاست؛ روش‌شناسی‌های بالا شامل چارچوب‌های اصلی به کار رفته در مطالعات تعامل انسان و کامپیوتر می‌شود. هر یک از این روش‌شناسی‌ها شامل اجزا و روش‌ها و تکنیک‌های گوناگونی برای ارزیابی، تحلیل، و طراحی رابط‌های کاربری کامپیوتری می‌شوند.

تعامل انسان و کامپیوتر (HCI) یک حوزه مطالعاتی بین‌رشته‌ای به نسبت جدید است که بر نحوه ارتباط برقرار کردن انسان‌ها با کامپیوتر برای انجام وظایف مختلف متمرکز است. در این مطالعات جنبه‌های گوناگون این تعامل از نقطه‌نگاه رشته‌های تخصصی مختلف مورد بررسی قرار می‌گیرد. محققان با درک درست نحوه ارتباط افراد و کاربران با سیستم‌های کامپیوتری در انجام وظایف توسط این سیستم‌ها، تلاش می‌کنند سیستم‌هایی با رابط‌های کاربری بهبود یافته ایجاد کنند که افراد به راحتی بتوانند با آنها ارتباط برقرار کنند.

این مطالعات بین‌رشته‌ای از جنبه‌های گوناگون و بر پایه فرضیه‌های متفاوتی به تعامل بین انسان‌ها و کامپیوترها می‌پردازد. در طول چند دهه تحقیقاتی که در این زمینه صورت گرفته، مجموعه‌ای از روش‌شناسی‌ها برای انجام این مطالعات ایجاد شده است. در این مطلب تلاش شد اصول کلی، اهداف، روش‌شناسی‌ها، و همچنین نحوه انجام این مطالعات معرفی شود. در نهایت شایان ذکر است که این بحث یکی از موضوعات پایه‌ای در علوم کامپیوتری است که بیشترین مباحث نظری در رشته کامپیوتر را به خود اختصاص می‌دهد. امیدواریم این مطلب مقدمه مناسبی برای تحقیقات بیشتر در این زمینه به خوانندگان ارائه داده باشد.

۲-۴ - تاریخچه طراحی وب سایت

در چند سال اخیر سبک‌های طراحی وب به شدت تغییر یافته است. وب سایت‌های ساده اولیه تنها حاوی متونی بودند که تنها تزئین آنها یک لینک ساده متنی آبی رنگ بود. این لینک‌های ساده ما را به سوی متون دیگر منتقل می‌کردند. از آن روزهای اولیه تا به امروز، طراحی وب یک سفر طولانی و بسیار پیچیده را شاهد بوده است که در ادامه سیری در تاریخ طراحی وب را با یکدیگر انجام خواهیم داد.

۲-۴-۱ - یک آغاز نوستالژیک

در اوایل دهه ۱۹۹۰ ما شاهد تولد اولین زبان نشر وب بودیم . بله ما داریم از Hyper Text Markup Language یا همان HTML ساده و دوستداشتنی آن روزهای دنیای طراحی وب (و البته دهه های بعدی) سخن می گوئیم.

این زبان مناسب اشتراک گذاری متون ساده و تصاویر سبک بود . مشاهده متون ساده با لینک های آبی رنگ ، آن روزها لذت عجیبی را برای وب گردان فراهم می آورد . شاید این دوران برای همیشه نوستالژیک ترین دوران وب باقی بماند.

۲-۴-۲ - جداول

در اواسط دهه ۱۹۹۰ ، طراحان به اهمیت وب پی بردند و به توسعه آن کمک بیشتری نمودند . در این زمان بود که رابط کاربری گرافیکی (GUI) به طراحان این امکان را داد تا تصاویر بیشتری را به همراه آیکون های گرافیکی به وبسایت ها اضافه کنند . این دوران همزمان با افزایش محبوبیت وب در بین مردم بود و وب توانسته بود خود را به عنوان ابزاری برای انتقال اطلاعات مطرح کند.

طراحان ، جداول را به عنوان ابزاری برای تنظیم متن و تصاویر گرافیکی خود به کار می گرفتند . قبل از طراحی جداول به عنوان یک ساختار طراحی صفحات وب هیچ راهی برای دسته بندی و بهبود نمایش محتواهای وب وجود نداشت . جداول به طراحان وب این امکان را می داد که ساختار مناسب تری برای نمایش متون و تصاویر به کاربران خود را در اختیار داشته باشند.

۳-۴-۲ - طراحی Flash

هنوز دهه ۹۰ میلادی خاتمه پذیرفته بود که یک تکنولوژی جدید پایه عرصه وجود گذاشت . بله داریم از **فلش (Flash)** صحبت می کنیم.

فلش یک پلتفرم نرم افزاری بود که طراحان را قادر می ساخت تا موسیقی ، ویدئو ، انیمیشن ، فایل های گیف را در وبسایتهاشان به راحتی منتشر کنند و تجربه کاربری (UX) بهتری را بوجود آورند . فلش همچنین باعث شد تا طراحان آزادی بیشتری را برای ایجاد وبسایت های تعاملی داشته باشند . در این دوران بیشترین خلاقیت در طراحی وب و پیشرفت تکنولوژیکی تا آن زمان بوجود آمد.

با کمک فلش برای اولین بار منوهای تعاملی ، انیمیشن های تزئینی و دکمه های حبابی ، که زیبایی های خاصی را به صفحات وب می دادند پا به عرصه وجود گذاشتند و این باعث جلب توجه بیشتر مردم به وب می شد . این درحالی بود که وب برای بسیاری از مردم هنوز مفهوم جدیدی بود و آن ها از این طراحی های بصری هیجان زده می شدند .

اما محبوبیت فلش بسیار کوتاه بود چراکه کاربران مجبور به بروزرسانی های مستمر پلاگین فلش نصب شده بر روی کامپیوترهای خود بودند . همچنین فلش برای کاربران محدودیت هایی را در استفاده و دسترسی به وبسایت ها بوجود می آورد .

۲_۴_۴ - هرکس یک طراح وب

اگر چه فلش نتوانست انتظارات را به طور کامل برآورده کند اما سلیقه و نظر طراحان وب و کاربران را دچار تحوّل کرد . دیگر هیچ کسی انتظار صفحات وب ساده را نداشت و همه منتظر یک وب متفاوت بودند . همزمان با ظهور اولین نسل های رسانه های اجتماعی شاهد تولد Cascading Style Sheets یا همان CSS دوستداشتنی خودمان بودیم .

CSS بهترین ابزار برای تعیین یک سبک خاص بدون نیاز به کدنویسی های پیچیده بود . ایده اصلی CSS این بود که محتوا (HTML) را از نحوه ارائه (CSS) جدا می کرد . این موضوع باعث می شد که طراحی یک وبسایت کاری تخصصی و پیچیده ای نباشد و هرکسی بتواند طراح سایت خود باشد .

در این زمان بود که قالب های مختلف طراحی و توسعه پیدا کردند و طراحان بسیاری به دنبال ارائه جذاب ترین قالب ها برای سلايق مختلف مشتریان خود بودند . قابليت استفاده مجدد در قالب ها باعث کاهش هزينه های طراحی ، قابليت دسترسي ساده و زیبایی بیشتر می شد.

۲_۴_۵ - طراحی تخت

در سال ۲۰۱۰ روند طراحی صفحات وب تغييرات اساسی پیدا کرد . چرا که در این زمان صفحات وب با ابزارهای متفاوتی نظیر گوشی های هوشمند و تبلت ها هم بازديد می شدند . این موضوع یک اصطلاح خاص را در طراحی بوجود آورد که تا آن زمان کمتر شنیده شده بود . آن اصطلاح عبارت بود از واکنش گرایی یا همان پاسخگو بودن (Responsive) که عنوان می کرد محتوا باید توانایی قرار گیری در هر ظرفی را داشته باشد.

واکنش گرایی یک رویکرد جدیدی را ایجاد می کرد که باعث استفاده متفاوتی از HTML و CSS در کنار یکدیگر می گردید . ایده اصلی واکنش گرایی این می باشد که وبسایت باید قابليت ارائه در محیط هایی با صفحات نمایش مختلف را داشته باشد . این ایده باعث افزایش بهره وری در طراحی ، توسعه و نگهداری صفحات وب می شود.

این موج طراحی صفحات وب به طراحی تخت (Flat Design) مشهور گردید . در طراحی تخت دو هدف کارآمدی و چشم نوازبودن مورد توجه قرار گرفت . امروزه طراحی تخت همچنان مورد توجه طراحان و کاربران می باشد . در این نوع طراحی اولویت ویژه ای به محتوا داده شده است . همچنین سعی شده است که بین محتوا ها ارتباطات قوی و موثری برقرار شود تا دسترسي کاربر به سادگی صورت پذیرد . از دیگر سو پیچیدگی های اضافه در طراحی صفحات وب به شکل چشمگیری کاهش یافته است.

۲_۴_۶ - آینده وب

هرچند تاریخ طراحی صفحات وب کوتاه می باشد اما در همین دوران کوتاه شاهد اعصار گوناگونی در تاریخچه صفحات وب بوده ایم . امروزه طراحان وب با این سوال اساسی مواجه اند که در آینده نزدیک چه باید بکنیم ؟ آنها به دنبال طراحی هایی هستند که تجربه کاربری بهتری را برای کاربران به ارمغان آورد . یک تجربه کاربری (UX) خوب ، تنها با کمک یک واسط کاربری (UI) خوب بوجود خواهد آمد. قسمت جالب توجه این است که این روش که امروزه توسط بسیاری از کشور ها و برنامه نویسان در دنیا استفاده می شود، تا حد زیادی شبیه به اولین مراحل رشد طراحی وب می باشد.

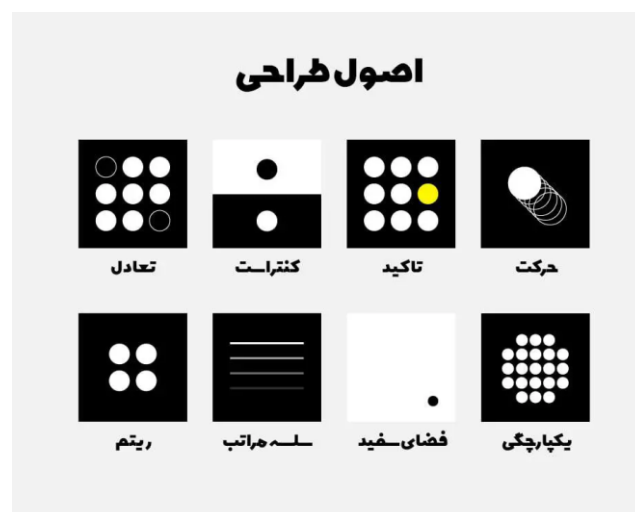
فصل سوم: اصول و چهارچوب های طراحی وب

۱-۳ - اصول طراحی

دانستن اینکه چه المان هایی باعث می شوند سایت شما طراحی خوبی داشته باشد و همچنین شما چگونه می توانید از آن ها در کار خودتان استفاده کنید مهم ترین نکته ایست که در این بخش به آن می پردازیم.

اصول اولیه ای از طراحی سایت، بر اساس نظریه ها و تجربیات هنرمندان و طراحان، برای جانمایی بخش ها و المان ها در سایت تعریف شده است. استفاده از این اصول می تواند به طراحان کمک می کند تا سایتی با ظاهر بسیار مرتب و زیبا طراحی کنند.

البته باید گفت که هیچکدام از این اصول را نمی توان قانون خواند و بیشتر باید به آن ها به عنوان راهنما و منبع الهام نگاه کرد درست همانطور که پیکاسو می گوید "قوانین را مثل یک انسان حرفه ای یاد بگیرید تا بتوانید آن ها را مانند یک هنرمند بشکنید"، هرگاه شما پایه و اساس طراحی وب را یاد گرفتید بعد از آن می توانید در اجرای آن ها جزئیات را به دلخواه خود تغییر دهید.

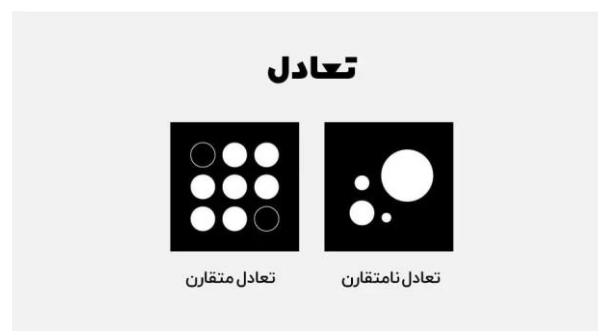


۳_۱_۱ - تعادل

تعادل بصری یکی از مهم‌ترین المان‌ها در طراحی وبسایت است یعنی ظاهر سایت باید طوری طراحی شود که فقط یک بخش جداگانه از سایت نتواند به تنهایی تمرکز کاربر را به خود جلب کند و باید کلیت بصری وبسایت باعث جذب مخاطب شود در طراحی سایت بهترین کار این است یک خط فرضی وسط صفحه وبسایت خود بکشید تا بتوانید تعادل را در آن رعایت کنید.

در طراحی وب دو نوع تعادل وجود دارد که در پایین به آن‌ها می‌پردازیم:

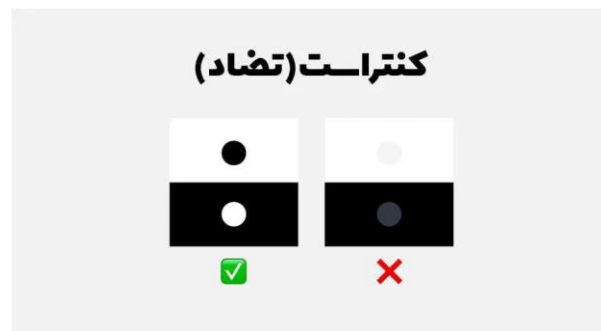
- تعادل متقارن زمانی است که دو طرف صفحه وبسایت مثل یک تصویر در آینه متقارن باشد و تعادلی داشته باشد که به کاربر حس نظم، مرتب بودن و زیبایی را القا کند.
- تعادل نامتقارن زمانی است که تعادل بصری هر دو طرف برابر باشد، اما ترکیب و ترتیب عناصر متفاوت باشد. طراحی نامتقارن متعادل یک رویکرد مدرن تلقی می‌شود که باعث می‌شود تا تجربه مخاطبان منحصر به فرد و خاص باشد.



۳_۱_۲ - کنتراست

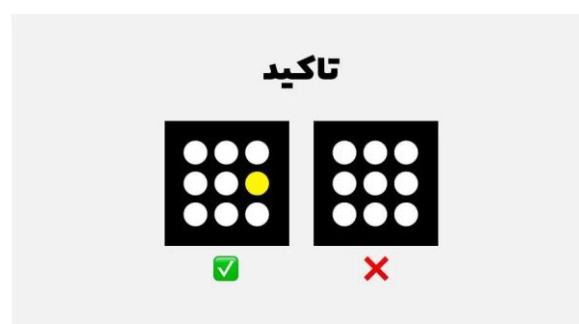
یکی المان‌هایی که باید در طراحی وب به آن توجه زیادی شود کنتراست است. کنتراست به این معناست که یک طراح وب بتواند تضادها را در کنار هم قرار دهد تا ظاهر سایت خود را بهتر نشان دهد. وقتی که کنتراست در

یک تصویر وجود داشته باشد باعث می‌شود تا به خاطر وجود تنوع و گیرایی بصری، مخاطبان سایت بازخورد بهتری نسبت به آن داشته باشند.



۳_۱_۳ - تاکید

یکی دیگر از المان‌های مهمی که باید در هنگام طراحی سایت حتماً به آن اهمیت زیادی بدهید تأکید است یعنی شما باید در طراحی سایت خود روی آن چیزی که تمرکز اصلی‌تان است حتماً تأکید کنید به طور مثال اگر تمرکز اصلی سایت شما تعمیرات موبایل است باید در صفحه اول خود بیشتر تأکید در طراحی خود را روی تعمیرات موبایل بگذارید. خواه به صورت عکس یا متن یا هر شکل دیگری سعی کنید تمرکز و تأکید بیشتری روی این موضوع داشته باشید.



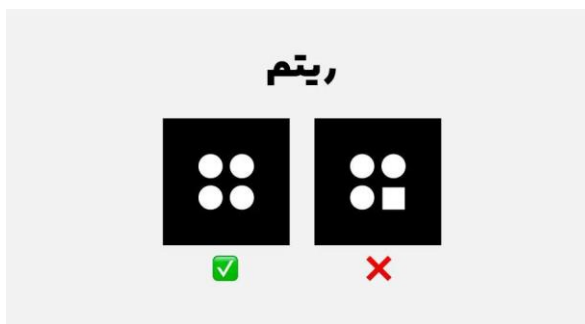
۳_۱_۴ - حرکت

یکی از نکاتی که می‌تواند تاثیر زیادی در موفقیت سایت داشته باشد حرکت است در واقع حرکت جزو قسمت‌های اساسی در طراحی وب است یک طراح وب می‌تواند با طراحی خوب حرکت چشم مخاطب را آن‌طور که مدنظرش است هدایت می‌کند و چیزهایی را که مهم هستند جلوی چشم مخاطب قرار دهد.



۳-۱-۵ - ریتم

منظور از ریتم تکرار عناصر مهم در صفحه اصلی است تا شما بتوانید در سایت خود انسجام ایجاد کنید. به طور مثال شما می‌توانید با یک کار ساده مانند استفاده از لوگویتان در تمام مطالب و یا استفاده از یک فونت یک دست، یک ریتم یکدست ایجاد کنید که همین یکپارچه بودن باعث می‌شود تا مخاطبتان خیلی راحت از وبسایت شما استفاده کند و از شلوغی و گمراهی آنها جلوگیری کنید.



۳-۱-۶ - سلسله مراتب

همانطور که در چیدن وسایل و مرتب کردن آنها، معمولاً از وسایل بزرگ‌تر در ابتدا شروع می‌کنیم، در طراحی سایت هم باید از سلسله مراتب استفاده کرد به طور مثال شما هیچ وقت نام کسب و کار خود را در پایین صفحه

وبسایتان قرار نمی‌دهید. بر همین اساس، پس از نام برند و لوگوی سایت تان، عنوان‌ها و موضوعات اصلی و کلی‌ای که در هر صفحه از سایت در خصوص آن قرار است صحبت کنید را به ترتیب ذکر کنید. شما در همان اول صفحه باید به مخاطب خود نشان دهید که در چه زمینه‌ای تخصص دارید.



۳_۱_۷ - فضای سفید

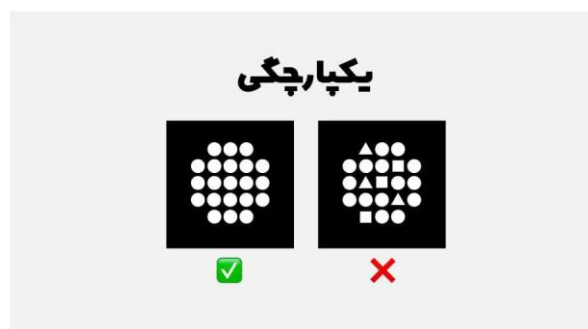
در طراحی هنری و جانمایی المان‌ها (کامپوزیشن) فضاهایی که خالی است را به عنوان فضای سفید می‌شناسند، فضای سفید در طراحی بصری وبسایت می‌تواند نقش بسیار مهمی را ایفا کند اگر شما تمام صفحه وبسایت خود را پر کنید، این نکته باعث می‌شود تا با ایجاد آلودگی بصری و شلوغی، مخاطب گیج شود و بلافاصله از سایت شما خارج شود پس باید بدانید که شما به عنوان یک طراح وبسایت نیاز دارید تا از فضای سفید به میزان کافی استفاده کنید تا وبسایتان شلوغ و بی هدف نباشد.



۳_۱_۸ - یک پارچگی

یک پارچگی زمانی اتفاق می افتد که تمام المان‌های دیگر که در بالا به آن‌ها اشاره کردیم در کنار هم قرار بگیرند و باعث ایجاد یک پارچگی شوند این یک پارچگی باعث می شود تا مخاطب به راحتی بتواند با وبسایت شما ارتباط برقرار کند.

ممکن است شما در شروع نتوانید در طراحی وبسایت خود یک پارچگی ایجاد کنید و نیاز داشته باشید تا چند بار این کار را انجام دهید تا بتوانید آن یک پارچگی مورد نیاز را به وجود آورید اما پس از به وجود آوردن یک پارچگی شما میفهمید که کدام المان‌ها در طراحی وبسایت شما ضرورت دارند، که این نکته باعث می شود تا چیزهایی که ضروری نیستند را حذف کنید تا بتوانید یک سایت یک پارچه که مورد توجه مخاطبان قرار می گیرد را داشته باشید.



۳_۲ - چیدمان وبسایت

چیدمان وبسایت حکم اسکلت سایت را دارد این چیدمان شماست که مدل بصری و همچنین نظم وبسایتتان را تعیین می کند شما در طراحی وبسایتان باید توجه بسیار زیادی به ظاهر آن داشته باشید چون برای ادامه راهتان بسیار ضروری و حیاتی است و به قولی صدای کسب و کار شما محسوب می شود و به جذب مخاطب برای شما کمک بسیار زیادی می کند.

چیدمان سایت شما توسط چند المان مهم تعیین می‌شود که یکی هدف وبسایتان است دیگری پیامی است که شما می‌خواهید به کاربرانان بدهید و در آخر محتوایی است که در وبسایتان تولید می‌کنید. با این که نمی‌توان قانون مشخصی برای چیدمان همه‌ی وبسایت‌ها تعیین کرد، اما این دو نوع چیدمان به صورت کلی بکارگرفته می‌شوند:

- **چیدمان مطابق با محتوا شما:** چیدمان وبسایت شما باید بر اساس محتوایی باشد که تولید می‌کنید

به طور مثال اگر شما قصد طراحی یک سایت فروشگاه‌ای را دارید تا بتوانید محصولات خود را به نمایش بگذارید بهتر است سایتتان را طوری طراحی کنید تا فضای زیادی برای برجسته کردن محصولاتتان وجود داشته باشد و اگر قصد طراحی یک وبلاگ را دارید باید طوری این وبلاگ را طراحی کنید تا بتوانید مطالبتان را به صورت سازمان یافته به مخاطبتان ارائه دهید.

- **چیدمان رایج:** در حال حاضر تعداد زیادی چیدمان وجود دارند که مدت‌هاست مورد استفاده قرار

گرفته‌اند و برای همه کسانی طراحی سایت انجام می‌دهند ثابت شده هستند این چیدمان‌های رایج برای کسانی که در طراحی سایت مبتدی هستند بسیار مناسب است و آن‌ها می‌توانند از آن‌ها برای شروع کار خود استفاده کنند.

عملکرد وبسایت شما اساساً به نحوه کار وبسایتان اشاره دارد. از سرعت و سهولت استفاده گرفته تا اقدامات خاصی که می‌توانید روی آن انجام دهید.

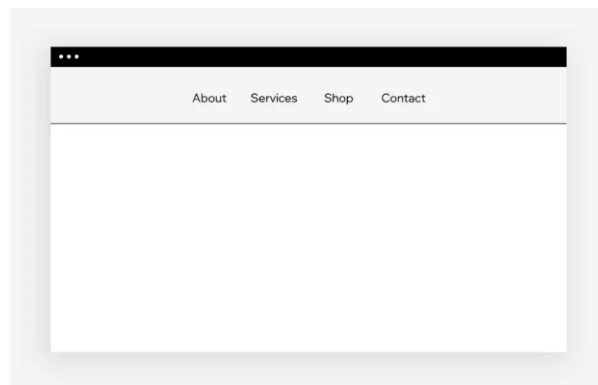
شرایط و امکانات وبسایت‌ها با توجه به تغییرات و پیشرفت‌هایی که در این حوزه و بطور کلی در اینترنت ایجاد شده، نسبت به گذشته کامل‌تر شده است.

۳_۳ - مسیریابی (نویگیشن)

همه وبسایت‌ها شامل چندین صفحه مختلف می‌شود اینجاست که شما در طراحی وبسایت‌تان باید دقت کنید که حتماً مسیریابی خوبی داشته باشید چون مخاطبان‌تان با مسیریابی در وبسایت شما می‌توانند به صفحه‌ای که می‌خواهند وارد شوند.

اضافه کردن یک منوی مسیریابی در وبسایت‌تان می‌تواند به مخاطبان‌تان کمک کند تا بتوانند راحت در سایت شما جستجو کنند و صفحه مورد نظرشان را پیدا کنند. بر اساس مدل وبسایت‌تان شما می‌توانید از منوهایی که به آن‌ها اشاره می‌کنیم استفاده کنید.

- **منوی مسیریابی کلاسیک:** این نوع منو در هدر سایت قرار دارد و به صورت یک لیست افقی است.

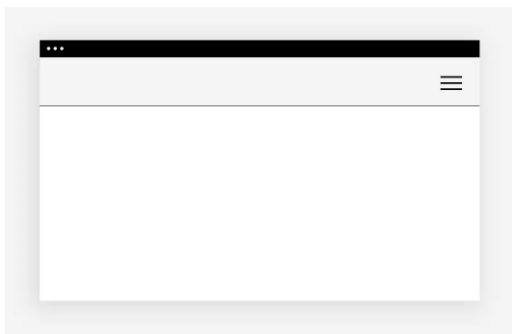


- **منوی چسبنده:** که به آن منوی شناور نیز گفته می‌شود در هنگام باسکرول کردن بازید کننده گان با

آن‌ها می‌ماند (به بالای صفحه چسبانده شده است).

- **منوی همبرگر:** آیکنی است که از سه نوار افقی تشکیل شده و پس از کلیک روی آن به طور کامل

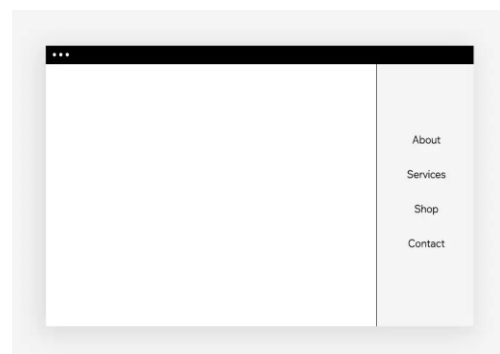
باز می‌شود.



- **منوی کشویی:** منویی است که بعد از کلیک روی یکی از موارد باز می شود و گزینه های مختلف را نشان می دهد.



- **منوی سایدبار:** لیستی از موارد است که در قسمت چپ یا راست سایت قرار دارد.



۳_۴ - UX

پس از این که در دهه هشتاد میلادی استفاده از کامپیوترهای خانگی به شدت افزایش یافت کمپانی‌های پیشرو در تکنولوژی به دنبال این بودند تا بفهمند که چگونه انسان می‌تواند با تکنولوژی بهتر ارتباط برقرار کند. UX. یا تجربه کاربری وقتی به درستی مورد استفاده قرار بگیرید می‌تواند تاثیر بزرگی در تجربه مخاطب داشته باشد.

پروژه طراحی UX برای بالا بردن کیفیت کلی وبسایت یعنی محتوای سایت، محصولات و تجربه کاربری توسط بهبود این هفت المان میسر می‌شود

- مفید بودن
- قابل استفاده بودن
- قابل کشف بودن
- معتبر بودن
- خواستنی بودن
- قابل دسترس بودن
- با ارزش بودن

۳_۵ - طراحی اداپتیو در مقابل طراحی ریسپانسیو

امروزه دستگاه‌های موبایل بیش از نیمی از ترافیک اینترنت را به خود اختصاص می‌دهند و ما به عنوان طراح وبسایت باید سایت‌های خودمان را با این دستگاه‌های کوچک تطبیق دهیم برای این که بتوانیم طراحی وبسایت خود را با موبایل تطبیق دهیم دو راهکار داریم یکی طراحی ریسپانسیو و دیگری طراحی اداپتیو. دانستن تفاوت

بین این دو روشن میتواند بسیار به کارمان بیاید مخصوصاً موقعی که می‌خواهیم پلتفرم طراحی سایتامان را مشخص کنیم:

- **طراحی اداپتیو:** در این مدل از طراحی شما باید برای هر پلتفرم یک سایت جداگانه طراحی کنید و آن را با پلتفرم مخصوص تطبیق دهید.

- **طراحی ریپانسیو:** در این مدل از طراحی سایت به شکل انعطاف پذیر طراحی می‌شود، در طراحی ریپانسیو سایت بر اساس دستگاهی که شما صفحه وبسایت مورد نظرتان را باز کرده‌اید خود را تغییر می‌دهد.

فصل چهارم: اجرا

۴-۱ - طرح اولیه و جمع آوری داده ها

ابتدا نمونه اولیه کد رو می سازیم و یا از تمپلیت ها و قالب های پیش ساخته با توجه به قواعد و چهارچوب های

فصل سوم استفاده می کنیم.

برای شروع، بخش های کلی را جمع آوری می کنیم. برای مثال در پروژه ما بخش های کلی که مشخص کننده

بخش های مختلف وب سایت هستند عبارتند از:

- هدر (در هدر این سایت نام و مشخصات فردی ذکر شده است)

- بخش شبکه های اجتماعی

- توضیحات فردی

- تحصیلات

- مهارت ها

- تجربیات

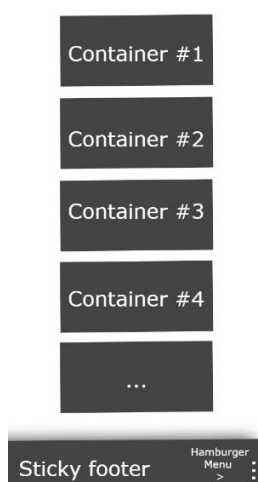
- مدارک

- رفرنس ها

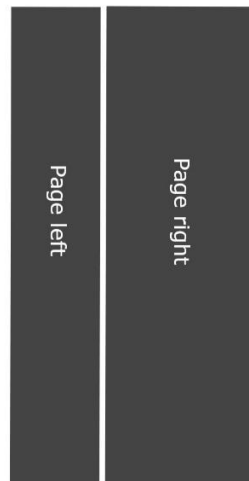
- زبان ها

- علاقه مندی ها

به هر کدام از بخش های بالا یک کانتینر اختصاص می دهیم. در ادامه سعی می کنیم شکل کلی وب سایت برای دستگاه های همراه را به شکل طرح اولیه بدست بیاوریم. برای طراحی وب سایت رزومه با اولیت واکنشگرایی، از آنجایی که نمایش مقدار زیادی از اطلاعات و بخش های مختلف وجود دارد، اولیت اولیه را برای طراحی و ایجاد یک صفحه کلی برای دستگاه های کوچک با رزولوشن پایین (عموماً گوشی ها) قرار می دهیم و شروع به طراحی می کنیم. با توجه به چهارچوب ها و قواعد فصل سوم شماتیک طراحی بدست آمده به شکل زیر است:



برای صفحات بزرگ هم به کانتینرها رو به دو بخش چپ و راست تقسیم می کنیم و تمامی پیچ به صورت دو سکشن چپ و راست تقسیم می شود که هر کدام دارای کانتینر های مربوطه اند و در داخل کد نیز به راحتی قابل تشخیص است.



برای شروع به کد زدن به ۳ فایل برای بخش های مختلف کد نیاز داریم. یک فایل برای کد های HTML، یک فایل برای کد های CSS و فایل برای جاوا اسکریپت.

نرم افزار ویرایش کد استفاده شده به نام Visual Studio Code پیشنهاد داده می شود. در ابتدا باید فایل ابتدایی و کد های ابتدایی هر کدام از زبان ها را ایجاد کنیم.

۲_۴ - کد ابتدایی HTML

کد ابتدایی (template) فایل html به نام index.html که آن را در ریشه (root) قرار می دهیم.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">

  <!--===== BOX ICONS =====-->
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/boxicons@latest/css/boxicons.min.css">

  <!--===== CSS =====-->
  <link rel="stylesheet" href="assets/css/styles.css">
```

```

    <title>Responsive resume cv</title>
</head>
<body>
    <!--===== HEADER =====>
    <header class="1-header" id="header">

</header>

    <main class="1-main bd-container">
        <!-- All elements within this div, is generated in PDF -->
        <div class="resume" id="area-cv">
            <div class="resume__left">
                <!--===== HOME =====>
                <section class="home" id="home">

                    <!-- Theme change button -->

                    <!-- Button to generate and download the pdf.
Available for desktop. -->

                </section>

                <!--===== SOCIAL =====>
                <section class="social section">

                </section>

                <!--===== PROFILE =====>
                <section class="profile section" id="profile">

                </section>

                <!--===== EDUCATION =====>
                <section class="education section" id="education">

                </section>

                <!--===== SKILLS =====>
                <section class="skills section" id="skills">

                </section>

            </div>

```

```

<div class="resume__right">
  <!--===== EXPERIENCE =====-->
  <section class="experience section" id="experience">

  </section>

  <!--===== CERTIFICATES =====-->
  <section class="certificate section" id="certificates">

  </section>

  <!--===== REFERENCES =====-->
  <section class="references section" id="references">

  </section>

  <!--===== LANGUAGES =====-->
  <section class="languages section">

  </section>

  <!--===== INTERESTS =====-->
  <section class="interests section">

  </section>
</div>
</div>
</main>

<!--===== SCROLL TOP =====-->

<!--===== HTML2PDF =====-->
<script src="assets/js/html2pdf.bundle.min.js"></script>

<!--===== MAIN JS =====-->
<script src="assets/js/main.js"></script>
</body>
</html>

```

نکات مهم:

- با استفاده از کامنت بخش های مختلف فایل برای فهم آسان تر کد جدا شده اند (خطوط سبز رنگ).

- برای هر کانتینر کلاس و سکشن آن را ایجاد می کنیم.
- آیکون های استفاده شده آنلاین هستند و آدرس آنها در ابتدای کد ذکر شده است.
- رابط فایل style.css را در فولدر css به آدرس assets/css/styles.css طبق کد بالا اضافه می کنیم.
- در انتهای کد فایل main.js را در فولدر js به آدرس assets/js/main.js طبق کد بالا اضافه می کنیم.

۴_۳ - کد ابتدایی CSS

کد ابتدایی (template) فایل CSS به نام styles.css که آن را در آدرس گفته شده قرار می دهیم.

```
/*===== GOOGLE FONTS =====*/
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&dis
lay=swap');

/*===== VARIABLES CSS =====*/
:root {
  --header-height: 3rem;

  /*===== Colors =====*/
  --title-color: #0B0A0A;
  --text-color: #403A3A;
  --text-color-light: #707070;
  --container-color: #FAFAFA;
  --container-color-alt: #F0EFEF;
  --body-color: #FCFCFC;

  /*===== Font and typography =====*/
  --body-font: 'Poppins', sans-serif;
  --h1-font-size: 1.5rem;
  --h2-font-size: 1.25rem;
  --h3-font-size: 1rem;
  --normal-font-size: .938rem;
  --small-font-size: .875rem;
```

```

--smaller-font-size: .813rem;

/*===== Font weight =====*/
--font-medium: 500;
--font-semi-bold: 600;

/*===== Margenes =====*/
--mb-1: .5rem;
--mb-2: 1rem;
--mb-3: 1.5rem;

/*===== z index =====*/
--z-tooltip: 10;
--z-fixed: 100;
}

/*===== BASE =====*/
/*===== Variables Dark theme =====*/
/*===== Button Dark/Light =====*/
/*===== Font size variables to scale cv =====*/
/*===== Generate PDF button =====*/
/*===== Classes modified to reduce size and print on A4 sheet
=====*/
/*===== CLASS CSS =====*/
/*===== LAYOUT =====*/
/*===== NAV =====*/
/* Show menu */
/* Active menu link */
/*===== HOME =====*/
/*===== SOCIAL =====*/
/*===== PROFILE =====*/
/*===== EDUCATION AND EXPERIENCE =====*/
/*===== SKILLS AND LANGUAGES =====*/
/*===== CERTIFICATES =====*/
/*===== REFERENCES =====*/
/*===== INTERESTS =====*/
/* Scroll top */
/*===== MEDIA QUERIES =====*/
/* For small devices, menu two columns */
/* Classes modified for large screen size */

```

کد ابتدایی (template) فایل JS به نام main.js که آن را در آدرس گفته شده قرار می‌دهیم.

```
/*===== SHOW MENU =====*/
/*===== REMOVE MENU MOBILE =====*/
/*===== SCROLL SECTIONS ACTIVE LINK =====*/
/*===== SHOW SCROLL TOP =====*/
/*===== DARK LIGHT THEME =====*/
/*===== REDUCE THE SIZE AND PRINT ON AN A4 SHEET =====*/
/*===== REMOVE THE SIZE WHEN THE CV IS DOWNLOADED =====*/
/*===== GENERATE PDF =====*/
// PDF generated area
// Html2pdf options
// Function to call areaCv and Html2Pdf options
// When the button is clicked, it executes the three functions
// 1. The class .scale-cv is added to the body, where it reduces
the size of the elements
// 2. The PDF is generated
// 3. The .scale-cv class is removed from the body after 5
seconds to return to normal size.
```

در حال حاضر ۳ فایل اصلی پروژه را ایجاد کردیم و برای کد نویسی دقیق برای بخش‌های متفاوت و کانتینر های مرتبط با هر بخش آماده هستیم.

۵_۴ – اضافه کردن آیکون‌ها، فونت‌ها و رنگ بندی

استفاده از سایت boxicon.com آیکون‌های مورد نظر خود را از طریق فایل CSS و یا HTML می‌توانیم به پروژه اضافه کنیم. در کنار هر آیکون متن مورد نظر برای اضافه کردن آیکن وجود دارد و براحتی با کلیک بر روی متن می‌توان آن را کپی کرد. همچنین با استفاده از fonts.google.com می‌توانیم فونت‌های مورد نظر خود را به پروژه اضافه کنیم. همچنین می‌توانیم کدهای مورد نظر برای رنگ بندی را انجام دهیم که در ادامه توضیح داده شده است:

```

<!--===== BOX ICONS(inside HTML) =====-->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/boxicons@latest/css/boxicons.min.css"
">

/*===== GOOGLE FONTS(inside CSS) =====*/
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;600&
display=swap');

/*===== VARIABLES =====*/
:root {
  --header-height: 3rem;

  /*===== Colors =====*/
  --title-color: #0B0A0A;
  --text-color: #403A3A;
  --text-color-light: #707070;
  --container-color: #FAFAFA;
  --container-color-alt: #F0EFEF;
  --body-color: #FCFCFC;

  /*===== Font and typography =====*/
  --body-font: 'Poppins', sans-serif;
  --h1-font-size: 1.5rem;
  --h2-font-size: 1.25rem;
  --h3-font-size: 1rem;
  --normal-font-size: .938rem;
  --small-font-size: .875rem;
  --smaller-font-size: .813rem;

  /*===== Font weight =====*/
  --font-medium: 500;
  --font-semi-bold: 600;

  /*===== Margenes =====*/
  --mb-1: .5rem;
  --mb-2: 1rem;
  --mb-3: 1.5rem;

  /*===== z index =====*/
  --z-tooltip: 10;
  --z-fixed: 100;
}

```



```

/*===== BASE =====*/
*,::before,::after{
  box-sizing: border-box;
}
html{
  scroll-behavior: smooth;
}
body{
  margin: 0 0 var(--header-height) 0;
  padding: 0;
  font-family: var(--body-font);
  font-size: var(--normal-font-size);
  background-color: var(--body-color);
  color: var(--text-color);
}
h1,h2,h3,ul,p{
  margin: 0;
}
h1,h2,h3{
  color: var(--title-color);
  font-weight: var(--font-medium);
}
ul{
  padding: 0;
  list-style: none;
}
a{
  text-decoration: none;
}
img{
  max-width: 100%;
  max-height: auto;
}
/*===== CLASS CSS =====*/
.section{
  padding: 1.5rem 0;
}

```

```
.section-title{
  font-size: var(--h2-font-size);
  color: var(--title-color);
  font-weight: var(--font-semi-bold);
  text-transform: uppercase;
  letter-spacing: .35rem;
  text-align: center;
  margin-bottom: var(--mb-3);
}
```

تا به اینجا ورودی های اصلی فایل های CSS و HTML را اضافه کردیم و می توانیم ساخت هدر و منوی چسبنده پایین صفحه را طبق طرح اولیه شروع کنیم.

۶-۴ - هدر و منوی نوپیشن

```
<!--===== HEADER =====-->
<header class="l-header" id="header">
  <nav class="nav bd-container">
    <a href="#" class="nav__logo">Mohammad Reza Nateghi</a>

    <div class="nav__menu" id="nav-menu">
      <ul class="nav__list">
        <li class="nav__item">
          <a href="#home" class="nav__link">
            <i class='bx bx-home
nav__icon'></i>Home
          </a>
        </li>

        <li class="nav__item">
          <a href="#profile" class="nav__link">
            <i class='bx bx-user' ></i>Profile
          </a>
        </li>

        <li class="nav__item">
```

```

        <a href="#education" class="nav__link">
            <i class='bx bx-book'></i>Education
        </a>
    </li>

    <li class="nav__item">
        <a href="#skills" class="nav__link">
            <i class='bx bx-receipt'></i>Skills
        </a>
    </li>

    <li class="nav__item">
        <a href="#experience" class="nav__link">
            <i class='bx bx-briefcase'
></i>Experience
        </a>
    </li>

    <li class="nav__item">
        <a href="#certificates" class="nav__link">
            <i class='bx bx-award'
></i>Certificates
        </a>
    </li>

    <li class="nav__item">
        <a href="#references" class="nav__link">
            <i class='bx bx-link-
external'></i>References
        </a>
    </li>
</ul>
</div>

<div class="nav__toggle" id="nav-toggle">
    <i class='bx bx-grid-alt' ></i>
</div>
</nav>
</header>

```

با افزودن کد بالا، تمامی منوهای نویگیشن و هدر اجرا شده (توجه شود که آیکون ها به روش توضیح داده شده از کلاس CSS آنلاین که اضافه کردیم به نام `boxicons.min.css` که در ابتدا به کد HTML اضافه کردیم) انشعاب گرفته شده است.

در ادامه به بخش CSS می رویم و بخش LAYOUT را تکمیل می کنیم. این بخش بسیار مهم است زیرا اولین بخش واکنش گرایی وب سایت است. می خواهیم در این بخش طراحی ابتدایی خود را اجرایی کنیم. بطوری که جای Header و منوی وب سایت بر اساس رزولوشن تغییر کند. برای دیوایس هایی با رزولوشن کمتر (عرض تصویر کمتر از ۹۶۸ پیکسل بین دو طرح سوییچ شود).

```
/*===== LAYOUT =====*/
.bd-container{
  max-width: 968px;
  width: calc(100% - 3rem);
  margin-left: var(--mb-3);
  margin-right: var(--mb-3);
}

.bd-grid{
  display: grid;
  gap: 1.5rem;
}

.l-header{
  width: 100%;
  position: fixed;
  bottom: 0;
  left: 0;
  z-index: var(--z-fixed);
  background-color: var(--body-color);
  box-shadow: 0 -1px 4px rgba(0,0,0,.1);
  transition: .3s;
}

/*===== NAV =====*/
.nav{
  height: var(--header-height);
  display: flex;
  justify-content: space-between;
```

```

align-items: center;
}

@media screen and (max-width: 968px){
  .nav__menu{
    position: fixed;
    bottom: -100%;
    left: 0;
    width: 100%;
    padding: 2rem 1.5rem;
    background-color: var(--body-color);
    box-shadow: 0 1px 4px rgba(0,0,0,.1);
    border-radius: 1rem 1rem 0 0;
    z-index: var(--z-fixed);
    transition: .3s;
  }
}

.nav__logo,
.nav__toggle{
  color: var(--title-color);
  font-weight: var(--font-medium);
}

.nav__toggle{
  font-size: 1.2rem;
  cursor: pointer;
}

.nav__item{
  text-align: center;
}

.nav__list{
  display: grid;
  grid-template-columns: repeat(3, 1fr);
  gap: 2rem;
}

.nav__link{
  display: flex;
  flex-direction: column;
  font-size: var(--smaller-font-size);
  color: var(--text-color-light);
  font-weight: var(--font-medium);
}

```

```

.nav__link:hover{
  color: var(--title-color);
}

.nav__icon{
  font-size: 1.2rem;
  margin-bottom: var(--mb-1);
}

/* Show menu */
.show-menu{
  bottom: var(--header-height);
}

/* Active menu link */
.active-link{
  color: var(--title-color);
}

```

در این بخش به سراغ جاوا اسکریپت رفته و تابع هایی برای صدا زدن و اجرای توابع درون فایل استایلز نیاز داریم

که به شکل زیر آنها را ایجاد می کنیم. (nav.toggle)

```

/*===== SHOW MENU (inside Main.JS) =====*/
const showMenu = (toggleId, navId) =>{
  const toggle = document.getElementById(toggleId),
  nav = document.getElementById(navId)

  // Validate that variables exist
  if(toggle && nav){
    toggle.addEventListener('click', ()=>{
      // We add the show-menu class to the div tag with the
      nav__menu class
      nav.classList.toggle('show-menu')
    })
  }
}
showMenu('nav-toggle', 'nav-menu')

```

و باز هم در ادامه لینک های درون منوی چسبنده (برای کاربران موبایل) را ایجاد می کنیم که به کانتینرهای متفاوت درون صفحه اشاره کند.

```
/*===== SCROLL SECTIONS ACTIVE LINK =====*/
const sections = document.querySelectorAll('section[id]')

function scrollActive(){
  const scrollY = window.pageYOffset

  sections.forEach(current =>{
    const sectionHeight = current.offsetHeight
    const sectionTop = current.offsetTop - 50;
    sectionId = current.getAttribute('id')

    if(scrollY > sectionTop && scrollY <= sectionTop +
sectionHeight){
      document.querySelector('.nav__menu a[href*=' + sectionId +
']').classList.add('active-link')
    }else{
      document.querySelector('.nav__menu a[href*=' + sectionId +
']').classList.remove('active-link')
    }
  })
}

window.addEventListener('scroll', scrollActive)
```

در این مرحله وب سایت می تواند سکشن خود را انتخاب کند و به سکشن مورد نظر (کلیک شده) اسکرول کند.

۷-۴ - طراحی کانتینر HOME

تمامی کانتینر ها درون کلاس main درون فایل html قرار دارند پس ابتدا کلاس main را می سازیم و سپس به ساخت کانتینر home می پردازیم.

```
<main class="l-main bd-container">
  <!-- All elements within this div, is generated in PDF -->
  <div class="resume" id="area-cv">
    <div class="resume__left">
```

```

<!--===== HOME =====-->
<section class="home" id="home">
    <div class="home__container section bd-grid">
        <div class="home__data bd-grid">
            

            <h1 class="home__title">Mohammad Reza
<b>Nateghi</b></h1>

            <h3 class="home__profession">IT
Specialist / Consultant</h3>

            <!-- Button to generate and download
the pdf. Available for desktop. -->
            <div>
                <a download=""
href="assets/pdf/ResumeCv.pdf" class="home__button-movil">Download</a>
            </div>

            <div class="home__address bd-grid">
                <span class="home__information">
                    <i class='bx bx-map home__icon'
></i> Tehran, Iran

                </span>
                <span class="home__information">
                    <i class='bx bx-envelope
home__icon' ></i> user@gmail.com

                </span>
                <span class="home__information">
                    <i class='bx bx-phone home__icon'
></i> +98-937

                </span>
            </div>
        </div>

        <!-- Theme change button -->
        <i class='bx bx-moon change-theme'
title="Theme" id="theme-button"></i>

        <!-- Button to generate and download the pdf.
Available for desktop. -->

        <i class="bx bx-download generate-pdf"
title="Generate PDF" id="resume-button"></i>
    </section>

```


تمامی بخش های سفید اطلاعات شخصی است و می تواند بر اساس نظر طراح برای کاربری مناسب تغییر کند.

در ابتدای تابع main تمامی کانتینرهایی که قرار است به صفحات با عرض بیشتر از ۹۶۸ پیکسل واکنش نشان بدهند و به سمت چپ صفحه منتقل شوند درون کلاس resume__left قرار دارند که در ابتدای کد مشخص شده است.

حال به بخش CSS کانتینر home می پردازیم:

```
/*===== HOME =====*/
.home{
  position: relative;
}

.home__container{
  gap: 3rem;
}

.home__data{
  gap: .5rem;
  text-align: center;
}

.home__img{
  width: 120px;
  height: 120px;
  border-radius: 50%;
  justify-self: center;
  margin-bottom: var(--mb-1);
}

.home__title{
  font-size: var(--h1-font-size);
}

.home__profession{
  font-size: var(--normal-font-size);
  margin: var(--mb-1);
}

.home__address{
  gap: 1rem;
}
```

```

}

.home__information{
  display: flex;
  align-items: center;
  font-size: var(--small-font-size);
}

.home__icon{
  font-size: 1.2rem;
  margin-right: .25rem;
}

.home__button-movil{
  display: inline-block;
  border: 2px solid var(--text-color);
  color: var(--title-color);
  padding: 1rem 2rem;
  border-radius: .25rem;
  transition: .3s;
  font-weight: var(--font-medium);
  margin-top: var(--mb-3);
}

.home__button-movil:hover{
  background-color: var(--text-color);
  color: var(--container-color);
}

```

با استفاده از CSS آیکون های بخش های مختلف home را به آنها مرتبط می سازیم که در کد بالا مشخص شده و آیکان ها نیز یکسان سازی شده.

۸_۴ – کانتینر Social

در ابتدا کدهای بخش html را کامل می کنیم (توجه کنید که تمامی کانتینر ها درون کلاس مین قرار گرفتند و با استفاده از تگ section از یکدیگر جدا شده اند):

```

<!--===== SOCIAL =====-->
<section class="social section">
  <h2 class="section-title">SOCIAL</h2>

  <div class="social__container bd-grid">
    <a href="Https://linkedin.com/in/Nateghi7" target="_blank"
class="social__link">
      <i class="bx bxl-linkedin-square social__icon"></i>
@Nateghi7
    </a>
    <a href="https://github.com/Nateghi7" target="_blank"
class="social__link">
      <i class="bx bxl-github social__icon"></i> @Nateghi7
    </a>
    <a href="https://t.me/btczen" target="" class="social__link">
      <i class="bx bxl-telegram social__icon"></i> @btczen</a>
  </div>
</section>

```

لینک های مربوط به شبکه های اجتماعی متفاوت را وارد می کنیم. آیکون های مورد نظر از هر شبکه اجتماعی را

از وب سایت ارایه شده می توان به راحتی جایگزین کرد و لینک های مربوط به هر شبکه را همچنین به روز

رسانی کرد. سپس به سراغ کد CSS برای این کانیتیر می رویم:

```

/*===== SOCIAL =====*/
.social__container{
  grid-template-columns: max-content;
  gap: 1rem;
}

.social__link{
  display: inline-flex;
  align-items: center;
  font-size: var(--small-font-size);
  color: var(--text-color);
}

.social__link:hover{
  color: var(--title-color);
}

.social__icon{

```

```
font-size: 1.2rem;
margin-right: .25rem;
}
```

۹_۴ – کانتینر Profile

در واقع این کانتینر نکته مهمی ندارد و بیشتر شبیه به متن در رابطه من (About me) در دیگر وب سایت ها

است. ابتدا کد html را کامل می کنیم:

```
<section class="profile section" id="profile">
  <h2 class="section">Profile</h2>

  <p class="profile__description">TEXT ABOUT ME</p>
</section>
```

سپس کد CSS را کامل می کنیم:

```
/*===== PROFILE =====*/
.profile,
.profile__description{
  text-align: center;
}
```

۱۰_۴ – کانتینر Education

در این کانتینر و همچنین کانتینر Experience برای نشان دادن زمان از دایره و خط در کنار دیگر نوشته ها

استفاده می کنیم که بیانگر تایم لاین (Time Line) در هنگام نمایش است. در کدهای زده شده به نوشته های

line و همچنین rounder دقت کنید. ابتدا کد HTML را می نویسیم و هر جا که لازم بود تغییرات سلیقه ای

مد نظر را اعمال می کنیم.

```
<!--===== EDUCATION =====>
<section class="education section" id="education">
  <h2 class="section-title">Education</h2>
```

```

<div class="education__container bd-grid">
  <div class="education__content">
    <div class="education__time">
      <span class="education__rounder"></span>
      <span class="education__line"></span>
    </div>

    <div class="education_data bd-grid">
      <h3 class="education__title">Title</h3>
      <span class="education__studies">Studies</span>
      <span class="education__year">20XX - 20XX</span>
    </div>
  </div>
</div>
</section>

```

و پس از آن طبق روال گذشته به سراغ کد CSS می رویم. توجه کنید که به دلیل وجود المان های تایم لاین که

بین این کانتینر و کانتینر Experience مشترک هستند، هر دو با هم نوشته شده و در کد ساده سازی صورت

گرفته است:

```

/*===== EDUCATION AND EXPERIENCE =====*/
.education__content,
.experience__content{
  display: flex;
}

.education__time,
.experience__time{
  padding-right: 1rem;
}

.education__rounder,
.experience__rounder{
  position: relative;
  display: block;
  width: 16px;
  height: 16px;
  background-color: var(--text-color-light);
  border-radius: 50%;
  margin-top: .25rem;
  margin-right: 0.5rem;
}

```

```
.education__line,
.experience__line{
  display: block;
  width: 2px;
  height: 110%;
  background-color: var(--text-color-light);
  transform: translate(7px, 0);
}

.education__data,
.experience__data{
  gap: .5rem;
}

.education__title,
.experience__title{
  font-size: var(--h3-font-size);
}

.education__studies,
.experience__company{
  font-size: var(--small-font-size);
  color: var(--title-color);
}

.education__year{
  font-size: var(--smaller-font-size);
}
```

۴_۱۱ – کانتینر Skills

این کانتینرها بطور مشترک با کانتینر Language در بخش CSS المان‌های مشترک دارند و علاوه بر اینکه ساده هستند بلکه مشابه یکدیگر نیز هستند. این کانتینر آخرین کانتینر از سری کانتینرهای سمت چپ تصویر می‌باشد. پس بعد از اتمام آن کلاس resume__left را می‌بندیم (برای صفحات با رزولوشن بالا) و کلاس resume__right شروع می‌شود. طبق روال ابتدا به تکمیل کد HTML می‌پردازیم:

```
<!--===== SKILLS =====>
<section class="skills section" id="skills">
  <h2 class="section-title">Skills</h2>
```

```

<div class="skills__content bd-grid">
  <ul class="skills__data">
    <li class="skills__name">
      <span class="skills__circle"></span> Data1
    </li>
    <li class="skills__name">
      <span class="skills__circle"></span> Data2
    </li>
    <!--</ul>-->

    <!--<ul class="skills__data">-->
      <li class="skills__name">
        <span class="skills__circle"></span> Data3
      </li>
      <li class="skills__name">
        <span class="skills__circle"></span> Data4
      </li>
    </ul>
  </div>
</section>
</div>

<div class="resume__right">

```

سپس به کد مشترک CSS بین کانتینرهای Skills و Languages می پردازیم:

```

/*===== SKILLS AND LANGUAGES =====*/
/* changed from repeat(4,1fr) ;*/
.skills__content{
grid-template-columns: repeat(1,1fr) ;
}

.languages__content{
  grid-template-columns: repeat(2, 1fr);
}

.languages__content,
.skills__content{
  gap: 0;
}
/* changed from h3; + no font size command*/
.skills__name{
  /*margin-bottom: -0.1rem;*/
  align-items: center;
}

```

```

}

.languages__name{
  display: flex;
  align-items: center;
  margin-bottom: var(--h3-font-size);
}

.skills__circle,
.languages__circle{
  display: inline-flex;
  width: 5px;
  height: 5px;
  background-color: var(--text-color);
  border-radius: 50%;
  margin-right: .75rem;
}

.skills__circle{
  margin-bottom: .2rem;
}

```

۱۲_۴ - کانتینر Experience

برای کوتاه شدن کد، مقداری از زیر مجموعه ها حذف شده اند و فقط یک زیر مجموعه از این کانتینر در نظر

گرفته شده است، در صورت نیاز می توانید با کپی کردن تعداد زیر مجموعه ها را افزایش داد. در ابتدا به کد

HTML می پردازیم:

```

<!--===== EXPERIENCE =====>
<section class="experience section" id="experience">
  <h2 class="section-title">Experience</h2>

  <div class="experience__container bd-grid">
    <div class="experience__content">
      <div class="experience__time">
        <span class="experience__rounder"></span>
        <span class="experience__line"></span>
      </div>

      <div class="experience__data bd-grid">
        <h3 class="experience__title">Title</h3>

```



```

        <span class="experience__company">From 20XX to 20XX |
@Company</span>
        <p class="experience__description">Description</p>
    </div>
</div>
</div>
</section>

```

کد CSS این بخش قبلاً بصورت مشترک با کانتینر Education نوشته شده است.

۱۳_۴ - کانتینر Certificates

کد فایل HTML بصورت ساده شده (تنها دو کلاس) به صورت زیر نوشته می شود:

```

<!--===== CERTIFICATES =====-->
<section class="certificate section" id="certificates">
    <h2 class="section-title">Certificates</h2>

    <div class="certificate__container bd-grid">
        <div class="certificate__content">
            <h3 class="certificate__title">Title1</h3>
            <p class="certificate__description">Description1.</p>
        </div>

        <div class="certificate__content">
            <h3 class="certificate__title">Title2</h3>
            <p class="certificate__description">Description2</p>
        </div>
    </div>
</section>

```

کد CSS برای کانتینر Certificate:

```

/*===== CERTIFICATES =====*/
.certificate__title{
    font-size: var(--h3-font-size);
    margin-bottom: var(--mb-1);
}

```

References کانتینر - ۱۴_۴

بخش HTML :

```
<!--===== REFERENCES =====>
<section class="references section" id="references">
  <h2 class="section-title">References</h2>

  <div class="references__container bd-grid">
    <div class="references__content bd-grid">
      <span class="references__subtitle">Professor</span>
      <h3 class="references__title">Dr. Seyed Ali Razavi
Ebrahimi</h3>
      <ul class="references__contact">
        <li>Phone: +98-21-224-58-313</li>
        <li>Cell: +98-912-394-1660</li>
        <li>Email: ali_razavi@pnu.ac.ir</li>
      </ul>
    </div>
  </div>
</section>
```

بخش CSS :

```
/*===== REFERENCES =====*/
.references__content{
  gap: .25rem;
}

.references__subtitle{
  color: var(--text-color-light);
}

.references__subtitle,
.references__content{
  font-size: var(--smaller-font-size);
}

.references__title{
  font-size: var(--h3-font-size);
}
```

۱۵_۴ - کانتینر Interests

این کانتینر آخرین کانتینر است پس باید تمامی کلاس های میانی و تگ های باز را ببندیم. بخش HTML:

```
<!--===== INTERESTS =====>
<section class="interests section">
  <h2 class="section-title">Interests</h2>

  <div class="interests__container bd-grid">

    <div class="interests content">
      <i class='bx bx-music interests__icon'></i>
      <span class="interests__name">Interest1</span>
    </div>

    <div class="interests content">
      <i class='bx bxs-basketball interests__icon'></i>
      <span class="interests__name">Interest2</span>
    </div>

  </div>
</section>

</div> <!--==class="resume__right"==--!>
</div> <!--==class="resume" id="area-cv"==--!>
</main>
```

کد بخش CSS برای کانتینر Interests:

```
/*===== INTERESTS =====*/
.interests__container{
  grid-template-columns: repeat(2 ,1fr);
  margin-top: var(--mb-1);
}

.interests__content{
  display: flex;
  flex-direction: column;
  align-items: center;
  gap: 0;
```

```

}

.interests__icon{
  font-size: 1.4rem;
  color: var(--text-color-light);
  margin-bottom: .25rem;
  vertical-align: -20%;
}

```

۱۶_۴ - کد منو و انیمیشن Scroll top

این کد تنها بر روی دستگاه های همراه و رزولوشن پایین ایجاد می شود و در فایل استایلز (CSS) جای دارد:

```

/* Scroll top */
.scrolltop{
  position: fixed;
  right: 1rem;
  bottom: -20%;
  display: flex;
  justify-content: center;
  align-items: center;
  padding: .3rem;
  background-color: var(--container-color-alt);
  border-radius: .4rem;
  z-index: var(--z-tooltip);
  transition: .4s;
  visibility: hidden;
}

.scrolltop__icon{
  font-size: 1.2rem;
  color: var(--text-color);
}

.show-scroll{
  visibility: visible;
  bottom: 5rem;
}

```

این منو تنها هنگامی ظاهر می شود که در ابتدای صفحه نباشیم (فضا برای بالا رفتن وجود داشته باشد).

کد بخش جاوا اسکریپت:

```

/*===== SHOW SCROLL TOP =====*/
function scrollTop(){
    const scrollTop = document.getElementById('scroll-top');
    // When the scroll is higher than 560 viewport height, add the
    show-scroll class to the a tag with the scroll-top class
    if(this.scrollY >= 200) scrollTop.classList.add('show-scroll');
    else scrollTop.classList.remove('show-scroll')
}
window.addEventListener('scroll', scrollTop)

```

۱۲_۴ – بخش Media Queries – تکمیل واکنشگرایی سایت

این بخش بر اساس ساینز نمایشگر کاربر وب سایت به دو بخش زیر تقسیم می شود:

- در بخش اول برای صفحات نمایش با عرض کوچکتر از ۳۲۰ پیکسل – در این بخش تمامی اطلاعاتی که در این صفحات نمایش جا نمی گیرند را بصورت جدول در آمده و ادامه متن را در سطر بعدی نمایش می دهیم.
- در بخش دوم برای صفحات نمایش با عرض بیشتر از ۹۶۸ پیکسل که بیشتر کامپیوترهای شخصی هستند، طبق طرح اولیه صفحه به دو بخش چپ و راست تقسیم شده و فضای اختصاص یافته به هر قسمت و با توجه به شکل کلی کانتینرهای هر بخش، با دقت و بر اساس ساینز متن تنظیم شده است.

```

/*===== MEDIA QUERIES =====*/
/* For small devices, menu two columns */
@media screen and (max-width: 320px){
    .nav__list{
        grid-template-columns: repeat(2, 1fr);
        gap:1rem .5rem;
    }
}

/* Classes modified for large screen size */
@media screen and (min-width: 968px){
    body{

```

```

    margin: 3rem 0;
}

.bd-container{
    margin-left: auto;
    margin-right: auto;
}

.l-header,
.scrolltop{
    display: none;
}

.resume{
    display: grid;
    grid-template-columns: 0.5fr 1fr;
    background-color: var(--container-color);
    box-shadow: 0 0 8px rgba(13,12,12,.15);
}

.resume__left{
    background-color: var(--container-color-alt);
}

.resume__left,
.resume__right{
    padding: 0 1.5rem;
}

.generate-pdf{
    display: inline-block;
}

.section-title,
.profile__description{
    text-align: center;
}

.home__container{
    gap: 1.5rem;
}

.home__button-movil{
    display: none;
}

```

```

.references__container{
  grid-template-columns: repeat(2, 1fr);
}

.languages__content{
  grid-template-columns: repeat(3,max-content);
  column-gap: 5rem;
  align-items: center;
}

.interests__container{
  grid-template-columns: repeat(4,max-content);
  column-gap: 3rem;
}

.experience__content{
  text-align: justify;
}
}

```

۱۸_۴ - مود های تصویر روز و شب (button Dark/Light)

کدهای مرتبط با بخش CSS (به ترتیب، رنگ بندی و ایجاد دکمه و سوییچ بین روز و شب):

```

/*===== Variables Dark theme =====*/
body.dark-theme{
  --title-color: #F2F2F2;
  --text-color: #BFBFBF;
  --container-color: #212121;
  --container-color-alt: #181616;
  --body-color: #2B2B2B;
}

/*===== Button Dark/Light =====*/
.change-theme{
  position: absolute;
  right: 0;
  top: 2.2rem;
  display: flex;
  color: var(--text-color);
  font-size: 1.2rem;
}

```

```

    cursor: pointer;
}

.change-theme:hover{
    color: var(--title-color);
}

```

بخش جاوا اسکریپت:

```

/*===== DARK LIGHT THEME =====*/
const themeButton = document.getElementById('theme-button')
const darkTheme = 'dark-theme'
const iconTheme = 'bx-sun'

// Previously selected topic (if user selected)
const selectedTheme = localStorage.getItem('selected-theme')
const selectedIcon = localStorage.getItem('selected-icon')

// We obtain the current theme that the interface has by validating the
dark-theme class
const getCurrentTheme = () =>
document.body.classList.contains(darkTheme) ? 'dark' : 'light'
const getCurrentIcon = () => themeButton.classList.contains(iconTheme)
? 'bx-moon' : 'bx-sun'

// We validate if the user previously chose a topic
if (selectedTheme) {
    // If the validation is fulfilled, we ask what the issue was to know
    if we activated or deactivated the dark
    document.body.classList[selectedTheme === 'dark' ? 'add' :
'remove'](darkTheme)
    themeButton.classList[selectedIcon === 'bx-moon' ? 'add' :
'remove'](iconTheme)
}

// Activate / deactivate the theme manually with the button
themeButton.addEventListener('click', () => {
    // Add or remove the dark / icon theme
    document.body.classList.toggle(darkTheme)
    themeButton.classList.toggle(iconTheme)
    // We save the theme and the current icon that the user chose
    localStorage.setItem('selected-theme', getCurrentTheme())
    localStorage.setItem('selected-icon', getCurrentIcon())
})

```


۱۹_۴ - نکته نهایی

بخش های اصلی وب سایت تکمیل شده است. ادامه کد موجود بر روی گیت هاب و پروژه در رابطه با تغییر سایز المان ها برای جایگیری در یک صفحه A4 و تبدیل وب سایت به یک فایل PDF است که با توجه به این نکته که ممکن است نیاز طراح متفاوت باشد و همچنین نیازمند به سرور (بک اند) است و باعث طولانی شدن بیش از حد سایز گزارش پروژه شود، در این بخش توضیح داده نشده است، ولی با این حال برای علاقه مندان از طریق وب سایت گیت هاب در دسترس است.

فصل پنجم: نتیجه گیری و نکات نهایی

تست واکنش گرایی بودن سایت

- برای تشخیص سایت واکنش گرا ساده ترین راه، وارد کردن آدرس سایت در مرورگر موبایل و تبلت است که در صورت مشاهده سایت متوجه قابلیت ریسپانسیو آن خواهید شد.
- در صورت عدم دسترسی به موبایل سایت مورد نظر را در لپ تاپ یا کامپیوتر وارد کنید و صفحه سایت خود را zoom in و zoom out کنید. اگر اندازه و چیدمان محتوای آن تغییر کرد در صورتی که اسکرول افقی در صفحه سایتان نمایان نشد، سایت شما از قابلیت ریسپانسیو بهره می برد.
- ابزار گوگل (موبایل فرندلی) با وارد کردن آدرس اینترنتی سایتان در سایت [mobile-friendly](#) می توانید برای تست سایت خود اقدام نمایید.
- همچنین در سایت [ivetesttooresponsl](#) می توانید سایت خود را در انواع نمایشگرهای مختلف تست کنید.

اهمیت طراحی سایت واکنش گرا چیست؟

در طراحی سایت، یکی از مهمترین نکاتی که می توان به آن اشاره کرد افزایش بازدیدکنندگان برای سایت است. قابلیت واکنش گرایی سایت برای جذب مخاطبان و کاربران بازدیدکننده سایت از اهمیت بالایی برخوردار می باشد، زیرا اگر سایت شما قابلیت ریسپانسیو را نداشته باشد، بازدید از سایت برای کاربران آزاردهنده خواهد بود.

واکنش گرا بودن سایت نسبت به نمایشگر دستگاه های هوشمند مختلف باعث می شود که ما در هنگام بازدید سایت، مطالب و محتوا را در اندازه و ابعاد صحیح در صفحه ببینیم و از زوم کردن و اسکرول کردن صفحه در جهت های مختلف پرهیزیم.

در سال ۲۰۱۹ میلادی گوگل واکنش گرا بودن سایت را یک اصل مهم برای تمامی وب سایت ها در نظر گرفت. اکثر سایت هایی که قابلیت واکنش گرا و ریسپانسیو را نداشتند از نتایج جستجو گوگل به شدت افت می کردند. به همین علت امروزه قابلیت ریسپانسیو بودن وب سایت ها در سئو بسیار مهم شده است.

۴. مزیت طراحی سایت واکنش گرا

۱. انعطاف پذیری بالا:

یکی از مزیت های طراحی سایت ریسپانسیو، انعطاف پذیر بودن سایت است که با انواع نمایشگر ها در سایزهای مختلف مانند موبایل، تبلت و لپ تاپ و ... سازگاری دارد و در نمایش انواع سایت در طیف وسیعی از مرورگر های جدید و قدیمی پیشرو است.

۲. بهتر شدن نتیجه سئو:

طراحی سایت ریسپانسیو از نکات بسیار مهم سئو برای افزایش رتبه سایت در گوگل است و سایت هایی که قابلیت واکنش گرایی را دارند در نتیجه جستجوی گوگل، امتیاز ویژه ای نسبت به سایت های غیر ریسپانسیو دارند و گوگل سایت های ریسپانسیو را در الویت قرار می دهد.

۳. صرفه جویی در هزینه:

قبل از اینکه ویژگی واکنش گرا بودن سایت به صورت گسترده مورد استفاده قرار گیرد، کاربرانی که قصد داشتند سایتشان در مرورگرهای موبایل به درستی نمایش داده شود باید هزینه ای را جهت راه اندازی و طراحی مجدد سایت برای ساینز انواع نمایشگرها پرداخت می کردند. اما امروزه با قابلیت ریسپانسیو، دیگر نیازی به پرداخت هزینه های سنگین طراحی مجدد سایت نخواهید داشت.

۴. افزایش کاربران موبایل:

امروزه بیش از ۴۵ درصد از کاربران اینترنت، با تلفن همراه خود در بستر اینترنت هستند. اگر شما سایتی را که راه اندازی کرده اید در نمایشگرهای موبایل به خوبی نمایش داده نشود، بسیاری از مخاطبانی را که می توانید داشته باشید را از دست خواهید داد. قابلیت ریسپانسیو سایت سبب افزایش کاربران در انواع مرورگرهای موبایل می شود. زیرا با دسترسی راحت اینترنت در تلفن همراه، اکثر مخاطبان با موبایل همراه خود به وب گردی می پردازند.

منابع:

1. <https://www.hamyarit.com/basic/hci/>
2. <https://dangoweb.ir/%D8%B7%D8%B1%D8%A7%D8%AD%DB%8C-%D9%88%D8%A8-%DA%86%DB%8C%D8%B3%D8%AA/>
3. <https://www.irandnn.ir/mag/what-is-website-design/>
4. <https://taop.ir/article/80/%D8%AA%D8%A7%D8%B1%DB%8C%D8%AE%DA%86%D9%87-%D8%B7%D8%B1%D8%A7%D8%AD%DB%8C-%D9%88%D8%A8%D8%B3%D8%A7%DB%8C%D8%AA>
5. <https://webzi.ir/blog/starting/P152-web-design.html>
6. <https://github.com/bedimcode/responsive-resume-cv-smith>
7. <http://moaveni.ir/%D8%AA%D8%A7%D8%B1%DB%8C%D8%AE%DA%86%D9%87-%D8%B7%D8%B1%D8%A7%D8%AD%DB%8C-%D8%B5%D9%81%D8%AD%D8%A7%D8%AA-%D9%88%D8%A8/>
8. <https://www.elitedesign.ir/blog/4-features-of-responsive-design/>



Faculty of Engineering

Department of Computer

B.Sc. Final Project Report

Title of the Report:

Design and execution of a practical responsive website

Under Supervision of:

Professor Razavi Ebrahimi

By:

Mohammad Reza Nateghi

Winter

