سوالات بخش فرانت اند :

1 . به‌جای آن از ترکیب HTML، CSS، و JavaScript ساده بهره برده‌ایم . در ادامه توضیح می‌دهم چرا ممکن است این انتخاب صورت گرفته باشد:

HTMLو CSS ساده:

سادگی و کنترل کامل بر استایل‌ها و ساختار :استفاده از CSS خالص) بدون فریمورک‌های CSSمثل (Bootstrap به شما امکان می‌دهد کنترل کاملی بر طراحی و استایل عناصر داشته باشید و از کدهای اضافی و غیرضروری اجتناب کنید.

بهینه‌سازی برای پروژه‌های کوچک:
وقتی پروژه کوچک است یا الزامی به استایل پیچیده وجود ندارد، CSSساده سریع‌تر و بهینه‌تر است.

2 . ریسپانسیو بودن رابط کاربری به‌صورت جزئی و با استفاده از یک media queryبرای دستگاه‌های کوچک‌تر (با عرض کمتر از 480 پیکسل) پیاده‌سازی شده است.

3 . •از برچسب‌های استاندارد HTML مانند <form>، <input>، و <button> استفاده کرده‌اید که کمک می‌کند فناوری‌های کمکی مانند صفحه‌خوان‌ها (Screen Readers) ساختار صفحه را به‌درستی تفسیر کنند.

برچسب‌های جایگزین:(Placeholder)

فیلدهای ورودی (مانند شماره تماس و رمز عبور) دارای placeholderهستند که کاربران را در مورد اطلاعات موردنیاز راهنمایی می‌کنند. این کار مفید است، اما جایگزین مناسبی برای برچسب‌های مرتبط (<label>)نیست (توضیحات در بخش بهبودها.(

4 .

•شما از CSS و JavaScript ساده و بدون وابستگی به کتابخانه‌ها یا فریمورک‌های سنگین استفاده کرده‌اید. این امر باعث کاهش حجم کلی فایل‌ها و سرعت بالاتر بارگذاری می‌شود.

استفاده از یک فایل:HTML

تمام کدهای CSS و JavaScript درون همان فایل HTML قرار داده شده‌اند، که می‌تواند برای پروژه‌های کوچک مناسب باشد و باعث کاهش تعداد درخواست‌های HTTP شود.

عدم استفاده از تصاویر سنگین:

کد شما هیچ تصویری ندارد، که باعث کاهش زمان بارگذاری صفحه می‌شود.

. 5

ساختار ساده:

کد CSS و HTML در فایل واحدی قرار دارد، که برای پروژه‌های کوچک مناسب است. اما در پروژه‌های بزرگ‌تر این ساختار نگهداری و تغییرات را دشوار می‌کند.

کلاس‌های عمومی:

کلاس‌ها به طور عمومی و توصیفی تعریف شده‌اند (مانند `form-container`، `toggle-text`). این روش خوانایی را برای پروژه‌های کوچک حفظ می‌کند.

سوالات بخش بک اند :

. 1

زبان و فریمورک انتخاب‌شده Node.js :باExpress.js ، یکپارچگی بین فرانت‌اند و بک‌اند : از آنجا که جاوااسکریپت هم در فرانت‌اند و هم در بک‌اند استفاده می‌شود، انتخاب Node.js به تیم اجازه می‌دهد تا از یک زبان برای کل پروژه استفاده کند و دانش موجود را به‌طور کامل بهره‌برداری کند.

. 2

جلوگیری از :XSS استفاده از JSON برای ارسال داده‌ها و پاک‌سازی ورودی‌ها (Input Sanitization).
جلوگیری از :SQL Injection داده‌ها مستقیم به پایگاه داده ارسال نمی‌شوند، اما اگر از پایگاه داده استفاده شود، حتماً ازORM ها یا پارامترهای آماده (Prepared Statements) استفاده می‌کنیم.
مدیریت رمزهای عبور :ذخیره نکردن رمزها به‌صورت خام) در این پروژه فعلی ذخیره مستقیم است که نیاز به هش کردن با کتابخانه‌هایی مثل bcrypt دارد.(
:CORSمحدود کردن دسترسی بهAPI ها فقط برای دامنه‌های خاص.
توکن :JWT استفاده از توکن برای تایید هویت کاربران.

. 3

دیتابیس در این پروژه استفاده نشده است، و داده‌ها (مثل کاربران، محصولات و سبد خرید) به‌صورت استاتیک در آرایه‌ها ذخیره شده‌اند.

. 4

از معماری REST برای مدیریتAPI ها استفاده شده است .

ساختار:URL

APIها با استفاده ازURL های مشخص برای هر منبع (مثل /api/products برای محصولات، /api/register برای ثبت‌نام) طراحی شده‌اند.

. 5

مدیریت خطاهای سرور به روش‌های مختلف انجام شده است. برخی از روش‌های مدیریت خطا به شرح زیر است:

استفاده از کدهای وضعیت HTTP مناسب:

برای هر نوع خطا، کد وضعیت HTTP متناسب با نوع مشکل ارسال شده است:

400 Bad Request: برای درخواست‌های نادرست (مثلاً اگر ورودی معتبر نباشد یا داده‌ها به درستی ارسال نشده باشند).

404 Not Found: برای زمانی که منبع مورد نظر پیدا نشود (مثلاً وقتی کاربری با شماره تلفن مشخص وجود نداشته باشد).

201 Created: برای موفقیت در ثبت‌نام یا ایجاد داده جدید.

پاسخ‌های خطا در قالب:JSON

پیام‌های خطا به صورت JSONارسال می‌شوند، که معمولاً شامل یک پیام توضیحی از مشکل است (مثل " :message نام کاربری یافت نشد "یا :message "رمز اشتباه است ."(این کار باعث می‌شود که کلاینت‌ها بتوانند به راحتی خطاها را پردازش کنند.

سوالات مربوط به ارتباط فرانت‌اند و بک‌اند:

. 1

در این پروژه، ارتباط بین فرانت‌اند و بک‌اند از طریق APIهای REST برقرار شده است. فرانت‌اند از متدهای HTTP مانند GETو POSTبرای ارسال درخواست‌ها به سرور و دریافت پاسخ‌ها استفاده می‌کند. داده‌ها به فرمت JSONارسال و دریافت می‌شوند. برای نمونه:

ورود :فرانت‌اند با استفاده از POSTدرخواست ورود به مسیر /api/loginارسال می‌کند.

ثبت‌نام :درخواست ثبت‌نام با POSTبه /api/registerارسال می‌شود.

محصولات و سبد خرید :با استفاده از GETبه مسیرهای /api/productsو /api/cartدرخواست ارسال می‌شود.

این روش، ساده و مناسب برای ارتباط میان کلاینت و سرور است.

. 2

در کدی که ارائه داده‌اید، برای مدیریت stateدر فرانت‌اند از متغیرهای محلی و localStorageاستفاده شده است. توضیحات به شرح زیر:

مدیریت state در فرانت‌اند:

localStorage:برای ذخیره‌سازی اطلاعات کاربر (مثل توکن ورود) به کار رفته است. پس از ورود موفق، توکن دریافت شده از سرور در localStorageذخیره می‌شود و در درخواست‌های بعدی به سرور ارسال می‌گردد.

برای ذخیره‌سازی اطلاعات سبد خرید، از متغیر cartدر سرور استفاده شده که در حافظه موقت نگهداری می‌شود.

اتصال به بک‌اند:

از fetch APIبرای ارسال درخواست‌ها به سرور استفاده شده است.

درخواست‌ها به سرور از طریقURL های مختلف مانند /api/login, /api/register, /api/productsو /api/ordersارسال می‌شوند.

داده‌ها به صورت JSONبه سرور ارسال و پاسخ‌ها به صورت JSONدریافت می‌شوند.

در نتیجه، stateمدیریت می‌شود و ارتباط با بک‌اند از طریق درخواست‌های RESTful APIبرقرار می‌شود.

. 3

استفاده از توکن برای احراز هویت:

پس از ورود موفق، سرور یک توکن (JWT) به فرانت‌اند باز می‌گرداند و این توکن در localStorageذخیره می‌شود.

در درخواست‌های بعدی از این توکن برای احراز هویت و دسترسی به منابع حفاظت‌شده استفاده می‌شود) اگرچه این بخش به طور کامل در کد فعلی وجود ندارد، اما بر اساس الگوی معمول JWTقابل اجرا است.(

. 4

Authentication(احراز هویت) تا حدی پیاده‌سازی شده است، اما Authorization(مجوز دسترسی) به طور کامل پیاده‌سازی نشده است. جزئیات به شرح زیر است:

) Authenticationاحراز هویت:(

در روت ورود(`/api/login`) ، فرآیند احراز هویت انجام می‌شود. زمانی که کاربر شماره تلفن و رمز عبور را وارد می‌کند، سرور بررسی می‌کند که آیا اطلاعات وارد شده با اطلاعات کاربران موجود در سرور مطابقت دارد یا خیر.

سوالات کلی:

. 1

مهم‌ترین چالش‌ها و راه‌حل‌ها:

چالش مدیریت وضعیت :(state) استفاده از localStorageبرای ذخیره اطلاعات ورود و سبد خرید می‌تواند باعث مشکلاتی در مقیاس‌های بزرگتر شود. این مشکل با انتقال به مدیریت stateپیشرفته‌تر) مانند Redux یا (Context API قابل حل است.

چالش امنیت :ارسال اطلاعات حساس مانند رمز عبور به صورت متنی مشکل‌ساز است. برای حل این مشکل می‌توان از هش‌کردن رمز عبور در سرور و استفاده از HTTPS استفاده کرد.

. 2

تکنیک‌های تست و دیباگ:

استفاده از console.logبرای نمایش وضعیت متغیرها و پاسخ‌های.API

Network tabدر DevTools برای بررسی درخواست‌ها و پاسخ‌های API.

استفاده از postmanبرای تست درخواست‌های API به صورت مستقل از فرانت‌اند.

. 3

قابلیت توسعه پروژه:

پروژه قابلیت توسعه با افزودن امکاناتی مانند مدیریت سبد خرید پیشرفته، نظرات کاربران، پرداخت آنلاین و مدیریت کاربران را دارد.

استفاده از APIهای بیشتر و پیاده‌سازی Authorizationو Authenticationپیشرفته‌تر می‌تواند این پروژه را برای استفاده در مقیاس بزرگتر آماده کند.

5 . خیر