هادی ناظمی 9730034  
تمرین اول برنامه نویسی وب  
 سوال 1 )

1. http1.0 :

* ایجاد هدر برای request و response (HTTP version number, status code, content type)
* امنیت کمتر
* حتما لازم نیست ریسپانس به صورت فرامتن باشد (حتما نباید html باشد و میتواند شامل این موارد باشد . scripts, stylesheets, media
* بلافاصله پس از دریافت پاسخ خاتمه میابد.

1. http1.1 :

* در حال حاضر رایج است
* منابع را یکی پس از دیگری بارگیری می کند، بنابراین اگر یک منبع قابل بارگیری نباشد، همه منابع دیگر پشت آن را مسدود می کند.

1. Http2 :

* یک ارتباط TCP برای ارسال چندین stream از دیتا را استفاده میکند به طوری که هیچ منبعی هیچ منبع دیگری را مسدود نمی کند.multiplexing
* دیتا را به پیام های binary-code میشکند و این پیام هارا شماره گذاری میکند به طوری که کلاینت بداند هر پیام باینری متعلق به کدام جریان است.
* HTTP/2 به سرور اجازه می دهد تا قبل از درخواست کلاینت، محتوا را به یک کلاینت push کن تا در دنیای مدرن کاربردی تر باشد.
* HTTP/2 از روش فشرده سازی پیشرفته تری به نام HPACK استفاده می کند که اطلاعات اضافی را در بسته های هدر HTTP حذف می کنداین کار چند بایت را از هر بسته HTTP حذف می کند تا بارگذاری سریعتر انجام شود.

1. http3 : تفاوت اصلی http3 این است که به جای پروتکل لایه ی انتقال TCP ، روی QUIC اجرا میشود که سریعتر و امن تر است.

به این دلیل که udp کنترل تراکم و ازدحام را ندارد به اینخاطر سرعت ارسال بیشتر میشود چون بسته ها منتظر رسیدن بسته های دیگر نمیشوند.

سوال2 )

الف)

1. Stateless :

* سرور نیازی ندارد که از از وضعیت و تاریخچه ی کلاینت چیزی ذخیره کند تا اگر لازم شد بازیابی کند.
* طراحی سرور، پیاده سازی و معماری آن ساده است .
* مقیاسپذیری معماری سرور آسان است.
* هندل کردن crash ها ساده است .بعد از crash کردن سرور فقط نیازه که یکبار ریستارت بشود.

1. Stateful :

* نیاز است تا سرور اطلاعاتی درمورد session ذخیره کند.
* طراحی سرور، پیاده سازی و معماری آن سخت است .
* ممکنه بعد از crash کردن سرور کاربر مجبور باشد لاگ اوت کند و از اول کارش رو شروع کنه.
* Scaling در این معماری سخت است .

ب) stateless استفاده از کوکی ها سمت کاربر یا استفاده از jwt . کوکی ها اطلاعات کاربر را در سمت کلاینت ذخیره میکند و اگر نیاز به استفاده از اونها بود اطلاعات رو برای سرور میفرستد.

سوال3 ) الف) در مدل کلاینت سرور چون ارتباط فقط بین این دو طرف وجود دارد نمیتوان محتوایی که نیاز است تا برای کاربر بفرستیم، از قبل فرستاده شود و تازه هنگامی که کاربر ریکوست میفرستد آن محتوا از بخش های مختلف گرفته میشود و نمایش داده میشود که این کارایی را کاهش میدهد و در این مدل از api ها استفاده میشود. APIs enable robust integrations

در وبهوک بجای ارسال ریکوست ها توسط یک اپلیکیشن و گرفتن ریسپانس از دیگری که در مدل کلاینت سرور بود، هنگامی که یک event خاص اتفاق میافتد یک برنامه دیتا را به دیگر برنامه ها میفرستد.( **Webhooks offer lightweight data sharing**

Webhook ها گاهی اوقات به عنوان Reverse APIsنامیده می شوند، چون مقداری از مشخصات API را می دهند و شما باید یک API برای وب هوک طراحی کنید تا از آن استفاده کند. وب هوک یک درخواست HTTP برای برنامه شما ارسال می کند و مسئولیت تفسیر آن بر عهده ما است.

ب) باید در سمت سرور سناریو های مختلف رادرنظربگیریم .مثلا از subscribe ,publisher استفاده کنیم. یا اگر عملیاتی ناموفق بود دوباره به سرور اطلاع دهیم مثلا اگر یک عملیاتی با خطا مواجه شد پیام ناموفق بفرستیم .

سوال4 )



سوال5 )

1. 503, Service Unavailable 500 Internal Server Errorچون ارور در سمت بک اند اتفاق افتاده ممنکن است هندل کردن آن به صورت کلی اینترنال ارور گذاشته شده باشد.
2. 401 Unauthorized
3. 301 Moved Permanently و 404 not found زیرا صفحه قبلی یافت نشده و دامنه ی آن حذف شده.
4. 429 Too Many Requests
5. 200 OK
6. 403 Forbidden

سوال6 )

1. Reverse proxy
2. Forward proxy
3. reverse proxy