سوال ۱:

۱)

- HTTP 1:
  - متدهای GET، HEAD و POST.
  - ارتباط non-persistent.
  - وجود header برای request و response.

- HTTP 1.1:
  - متدهای اضافه‌تر مثل GET، HEAD، POST، PUT، DELETE و OPTIONS.
  - ارتباط persistent به صورت متوالی.
  - ارتباط طولانی‌تر نسبت به ورژن قبل.
  - ارسال دیتا به صورت chunk و مشکل Head-of-line-blocking.

- HTTP 2.0:
  - قابلیت multiplexing که ارتباط persistent به صورت موازی می‌شود.
  - ارسال دیتا به صورت Header-Frame و Data-frame و حل مشکل Head-of-line-blocking.
  - کل فرایند ریکوئست و ریسپانس در یک stream.
  - Server push.
  - Binary framing and pipelining.
  - TLS 1.2.

- HTTP 3:
  - پروتکل Quick UDP Internet Connection به جای TCP.
  - stream multiplexing and flow control.
  - TLS 1.3[1]

۲)

UDP پیام را به صورت broad cast ارسال می‌کند ولی TCP از روش Peer-to-Peer استفاده می‌کند.
این موضوع موجب استفاده از مزیت‌های پروتکل UDP می‌شود. برای مثال سرعت UDP خیلی از TCP بیش‌تر است زیرا UDP به صورت connectionless است و برای دریافت پاسخ صبر نمی‌کند.
پروتکل QUIC موجب استفاده از مزیت‌هایی که در نسخه‌ی قبل HTTP مورد استفاده نیستند می‌شود چون پروتکل UDP به کم شدن تاخیر(latency) افزایش سرعت چشم‌گیر و امنیت در ارتباطات ضعیف کمک می‌کند.

[1] https://ably.com/topic/http-2-vs-http-3

سوال ۲:

آ)

stateless: یک ارتباطی است که در آن هیچ اطلاعاتی توسط server ذخیره نمی‌شود و نیازی ندارد وضعیت را در طول درخواست‌های متعدد حفظ کند. داده‌ها توسط client به server ارسال می‌شود به گونه‌ای که هر بسته اطلاعات منتقل شده را می‌توان به صورت مجزا گرفت، بدون آن‌که اطلاعاتی از بسته‌های قبلی در ارتباط وجود داشته باشد. این خاصیت آن‌ها را برای برنامه‌های با حجم بالا ایده‌آل می‌سازد به این صورت که با حذف بار سرور ناشی از حفظ اطلاعات جلسه، عملکرد این برنامه‌ها را افزایش می‌دهد.

stateful: ارتباطی است که نیاز به نگه داشتن وضعیت داخلی روی server را دارد. برای مثال درTCP، هر دو سیستم در طول زندگی خود اطلاعات مربوط به خود جلسه را حفظ می‌کنند.[۳]

ب)

HTTP یک پروتکل stateless است و وضعیت را نگهداری نمی‌کند. برخی از برنامه‌ها نیاز دارند کاربرانشان را بشناسند و بر اساس آن پاسخ مناسب دهند.

برای حل این مشکل می‌توان از Cookie استفاده کرد. روند به این صورت است که client به server درخواست HTTP می‌دهد. اگر server قبلا آن client را ندیده باشد یک عدد شناسایی به نام cookie به client بر می‌گرداند و client درخواست‌های HTTP بعدی را با آن cookie می‌فرستد. این عدد، کلید شناسایی client در سمت server است و server اطلاعاتی را از درخواست‌های HTTP که با آن cookie فرستاده شده نگه‌داری می‌کند.

راه دیگر، روش token based است. در این روش بجای اینکه اطلاعاتی همچون نام کاربر و دسترسی‌های او و ... را پس از لاگین، به صورت session در سرور ذخیره کنیم، آن را با کمک یک کلید secret که فقط در سرور موجود است، رمزنگاری می‌کنیم و بجای session به کلاینت ارسال می‌کنیم. کلاینت برای ارتباطات بعدی، از طریق header درخواست یا حتی خود URL، توکن را ارسال می‌کند و در سمت سرور به سادگی با کلید secret بازگشایی می‌شود. این روش محدودیت‌های کوکی(مانند وابسته بودن به دامنه) را ندارد و از نظر امنیتی هم بهتر از session است.[۴]

---

[۲] https://www.geeksforgeeks.org/difference-between-stateless-and-stateful-protocol/
[۳] https://en.wikipedia.org/wiki/Stateless_protocol
[۴] https://9px.ir/note/all-methods-for-identifying-client-from-a-webpage/

سوال ۳:

آ)

در کاربردهای ذکر شده نیازمند ارتباط از سمت سرور به کلاینت هستیم ولی این حالت ممکن نیست زیرا در پروتکل HTTP کلاینت باید درخواست ارتباط را بدهد و سپس سرور پاسخ دهد. چالش اصلی شروع ارتباط توسط سرور است.

برای مثال یک الرت رو بخواهیم به کلاینت بدهیم وقتی ارتباطی وجود ندارد به مشکل مواجه می‌شویم.

ب)

- Long Polling: به طور خلاصه، کلاینت از سرور اطلاعات می خواهد. سرور هیچ داده ای ندارد و قبل از ارسال پاسخ، مدتی منتظر می ماند، اگر چیزی در طول انتظار ظاهر شود، سرور آن را ارسال می کند و درخواست را می‌بندد. اگر چیزی برای ارسال وجود نداشته باشد و حداکثر زمان انتظار به دست آید، سرور پاسخی ارسال می کند که داده ای وجود ندارد. در هر دو مورد، مشتری درخواست بعدی برای داده را باز می کند.
- WebSockets: این یک پروتکل ارتباطی است که کانال های ارتباطی full-duplex را از طریق یک اتصال TCP فراهم می کند.
- SSE: تفاوت اصلی با روش Polling این است که ما فقط یک اتصال دریافت می‌کنیم و جریان رویداد را از طریق آن ادامه می دهیم. Long Polling یک اتصال جدید برای هر سرکشی ایجاد می‌کند.[۵]

سوال ۴:

ss://asghar:1234!!@ss.myproxy.com:1234\#shadowSocks1

- نوع پروتکل(scheme): ss
- نام کاربری: asghar
- رمز عبور: ۱۲۳۴!!
- میزبان(host): ss.myproxy.com
- پورت: ۱۲۳۴
- fragment: #shadowSocks1

[۵] https://www.smashingmagazine.com/2018/02/sse-websockets-data-flow-http2/

سوال ۵:

۱) کد ۵۰۰ مناسب است. زیرا مشکل از سرور داخلی سایت یعنی سرور دیتابیس است که لاگین انجام نمی‌شود.

۲) کد ۴۰۱ مناسب است. زیرا اطلاعات برای گرفتن دسترسی غلط است.

۳) کد ۳۰۱ مناسب است. چون دامنه به طور کلی عوض و منتقل شده است.

۴) کد ۴۲۹ مناسب است. زیرا این کد برای دسترسی بیش‌از حد مجاز است.

۵) کد ۲۰۰ مناسب است. زیرا درخواست تمدید توکن موفقیت‌آمیز و OK است.

۶) کد ۴۰۳ مناسب است. زیرا کاربر اجازه دسترسی را ندارد.

سوال ۶:

۱): استفاده از forward proxy.

۲) استفاده از reverse proxy.

۳) استفاده از reverse proxy.