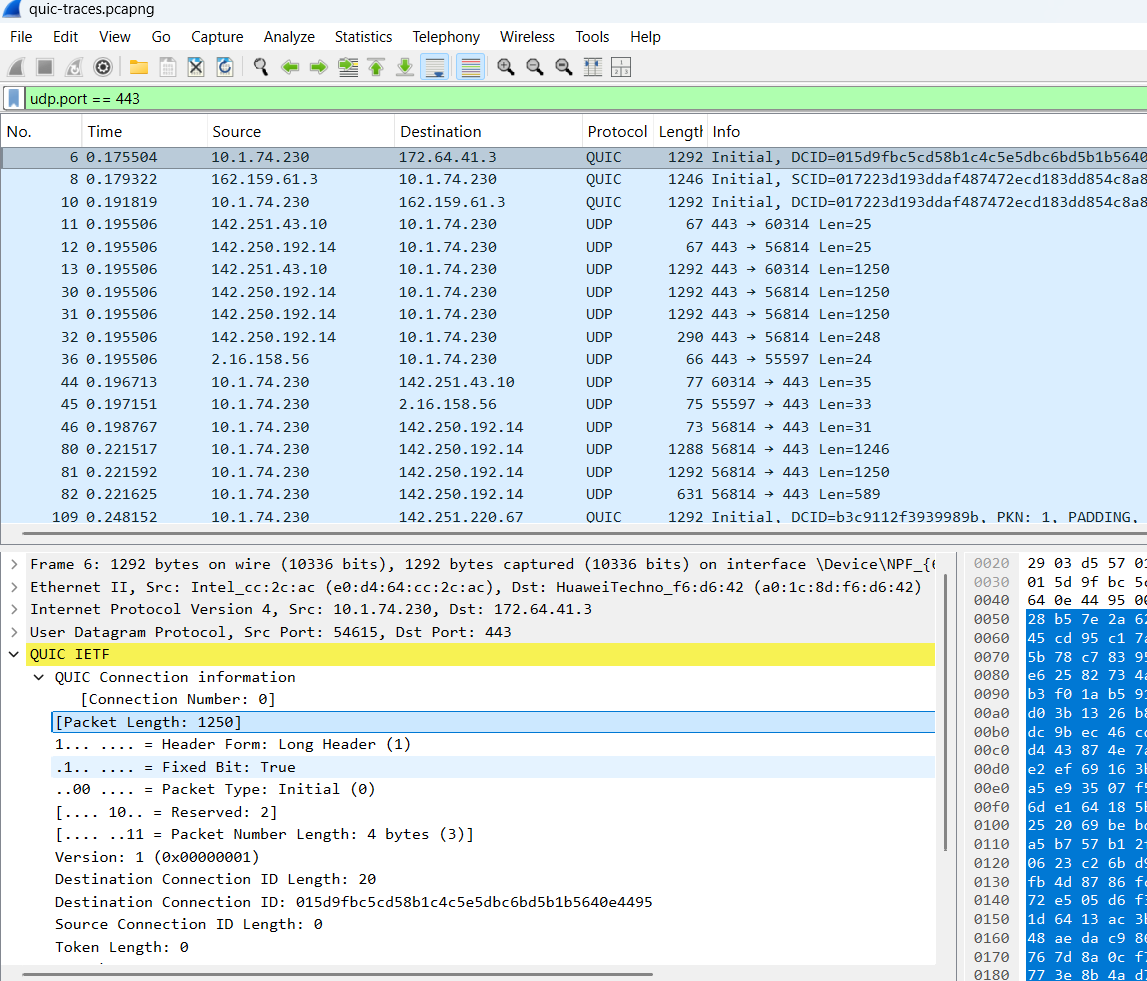
**Task 6:**

For the QUIC based website access, answer the following:

1. **What is the name of website?**

**solution:**

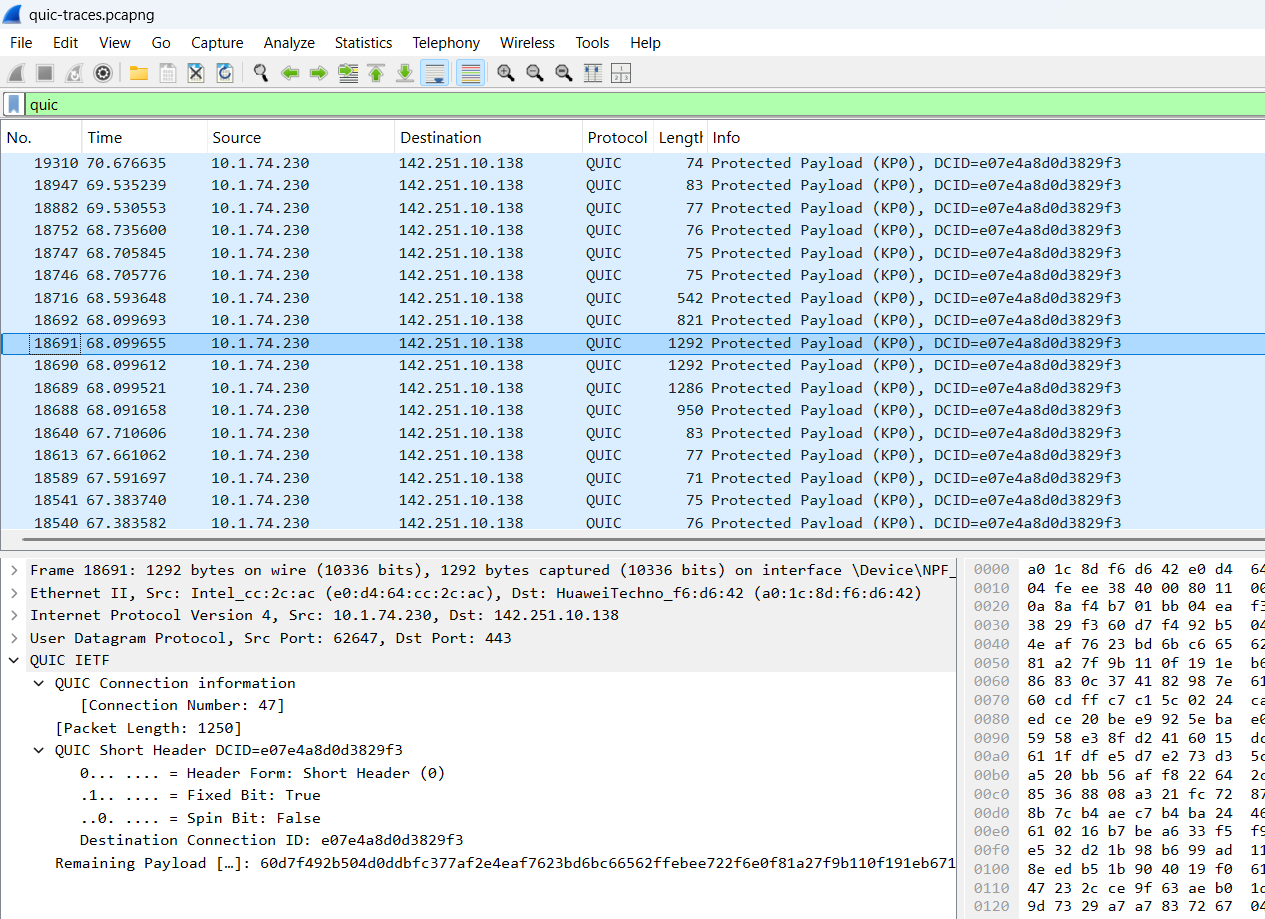
1.  Filter: quic **or** udp.port == 443.
2.  Open the first **Initial** packet → expand **QUIC → CRYPTO → TLS ClientHello** → find **SNI (server\_name)**.
3.  If decrypted, filter http3 and check :authority in the HTTP/3 request.
4.  **Record:** packet # and screenshot showing SNI or :authority.



**TASK:7**

**Packet containing the Initial QUIC handshake — what info is exchanged?**

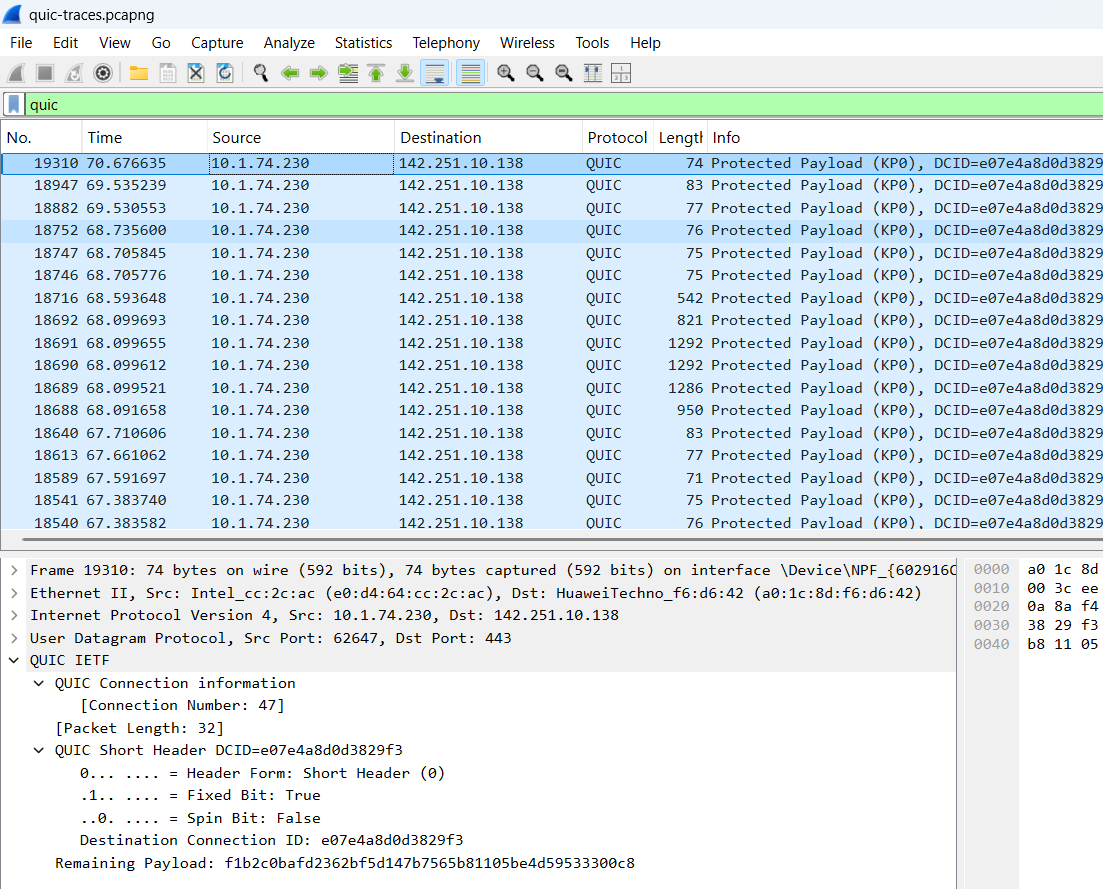
1. Filter: quic.
2. Find the first packet labelled **Long Header (Initial)** in the Info column.
3. Expand **QUIC header** and **CRYPTO** — note **Version**, **Destination/Source Connection IDs**, and embedded **TLS ClientHello** fields (cipher suites, extensions, key\_share, SNI).
4. **Record:** packet # and list those fields + screenshot.



**TASK:8**

**QUIC packet that contains TLS ClientHello**

1. Usually the same Initial packet(s). Filter quic and inspect Initial CRYPTO frames.
2. Expand **CRYPTO → TLSv1.3 Client Hello** (may be fragmented across Initial packets — check subsequent Initials).
3. **Record:** packet # (or range of packets if fragmented) and screenshot showing ClientHello (SNI, ALPN = h3, key\_share, supported\_versions).



**TASK:9**

**Which QUIC version is used?**

1. Open the **Initial** packet (Long Header) found above.
2. Expand the QUIC header → read the **Version** field (e.g., v1 / 0x00000001 or a draft number).
3. **Record:** packet # and exact Version string shown by Wireshark.

**TASK:10**

**5) Locate where 0-RTT or 1-RTT keys are first used**

1. After Initial + Handshake packets, look for packets with **Short Header** (or Info showing 0-RTT / 1-RTT). Filter quic and sort by time.
2. If you loaded SSLKEYLOGFILE, Wireshark will decrypt and show application frames — the first packet that Wireshark *decrypts* or shows “Short Header (decrypted)” is your 1-RTT. If client sent early data, you’ll see a client 0-RTT packet before handshake completes (marked 0-RTT).
3. **Record:** packet # and screenshot showing header type and the “decrypted / 0-RTT / 1-RTT” indicator.

**TASK:11**

**First packet carrying application data (HTTP/3) — difference from HTTP over TCP**

1. Filter: http3 (or quic and look for QUIC **STREAM** frames containing HTTP/3 frames).
2. The first packet showing http3.headers or http3.data is the first app-data packet. Note packet # and screenshot of headers (:method, :path, :authority).
3. Write the difference (short bullets):
   * HTTP/3 runs over **QUIC (UDP)** vs HTTP/1/2/HTTPS over **TCP**.
   * QUIC integrates **TLS1.3** (crypto in transport), reducing RTTs.
   * QUIC provides **multiplexed streams** at transport layer → **no TCP head-of-line blocking**.
   * HTTP/3 uses **QPACK** header compression vs HPACK (HTTP/2).