**🟢 Alternate Solutions in Tomcat (without Redis/DB)**

**1. Tomcat Session Replication (Cluster Mode)**

* Tomcat has a built-in **clustering** feature that replicates session data across all nodes.
* When one server (road) goes down, another server already has a copy of the cart (session).
* Configuration is done in server.xml:
* <Engine name="Catalina" defaultHost="localhost" jvmRoute="instance1">
* <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
* </Engine>
* ✅ **Pros:** No external system needed.
* ❌ **Cons:** More network traffic (sessions copied across all servers). Can become inefficient with many nodes.

**2. Persistent Session Store (Tomcat’s File or Database Persistence)**

* Tomcat can **save session data to disk or a database** when shutting down or when sessions are idle.
* This way, if a server restarts, it can reload sessions.
* Config via context.xml:
* <Manager className="org.apache.catalina.session.PersistentManager"
* saveOnRestart="true">
* <Store className="org.apache.catalina.session.FileStore"/>
* </Manager>
* ✅ **Pros:** Built into Tomcat, avoids external cache.
* ❌ **Cons:** Slower than memory replication, not great for large clusters.

**3. Stateless Design (Token/JWT Instead of Session)**

* Instead of storing sessions in Tomcat, switch the app to **JWT tokens** or client-side state.
* Apache just routes requests, and Tomcat doesn’t hold session data.
* ✅ **Pros:** No replication, no external DB. Very scalable.
* ❌ **Cons:** Requires **application code changes**.

**4. Hybrid: Sticky Sessions + Backup**

* Use sticky sessions with Apache, but also configure **session failover** in Tomcat cluster.
* Example: Most requests hit the same server, but if it fails, another one can recover the session from replication.

**🔑 ELI5 for Manager**

* Redis/DB is like a **shared cart room** → best practice.
* In Tomcat, we can do:
  1. **Copy carts to every road (session replication)** → simple, but lots of copying.
  2. **Save carts on disk/database (persistent manager)** → slower, but survives restarts.
  3. **No carts at all (stateless JWT tokens)** → users carry their own info.

👉 Do you want me to give you a **step-by-step Tomcat cluster setup guide with session replication** (the most common alternate fix), so you can try it in your environment?