# **Assignment: LDAP Authentication Wrapper with MFA Support**

## **🎯 Objective**

Design and implement a **Java-based LDAP authentication wrapper** that supports:

* **Primary authentication** via an enterprise LDAP directory (e.g., Active Directory, OpenLDAP).
* **Secondary authentication** using a **pluggable multi-factor authentication (MFA)** mechanism (specifically, OTP via email).
* **Secure** and **scalable** solution.

## **📦 Deliverables**

The candidate is expected to deliver:

1. **A Java library or wrapper module** that:
   1. Accepts authentication requests.
   2. Validates credentials against an LDAP server.
   3. Triggers and verifies secondary authentication via email OTP.
   4. Exposes clean and well-documented interfaces (e.g., REST API or Java SDK).
2. **A demo client or application** that:
   1. Demonstrates integration with the wrapper.
   2. Allows end-to-end authentication flow testing (LDAP login + MFA via email link or code).
3. **Documentation** that includes:
   1. Setup instructions (LDAP configuration, email service setup, how to run).
   2. Integration guide for clients.
   3. Explanation of security and performance considerations.
4. **Source Code** that follows:
   1. Clean, modular architecture (layered or hexagonal preferred).
   2. 90%+ unit test coverage.
   3. CI/CD ready (with static analysis and vulnerability checks).

## **✅ Functional Requirements**

### **1. LDAP Authentication Wrapper**

* Authenticate user credentials using Bind DN and search filters.
* Support for **multiple LDAP configurations** (e.g., failover, load balancing).
* Secure LDAP protocols: **LDAPS** and **StartTLS**.
* Connection management: configurable **timeouts**, **connection pooling**.

### **2. MFA Integration (Email OTP)**

* Pluggable architecture for MFA (allow adding new methods in future).
* **Email OTP flow**:
  + After LDAP login, send a one-time link/code to the user's email.
  + The authentication request should pause until the user clicks the link or enters the OTP.

### **3. Audit Logging & Alerts**

* Record all login attempts, MFA events, and failures.
* Alert or flag:
  + Brute-force login attempts.
  + Repeated OTP failures or unusual login patterns.

## **🔒 Non-Functional Requirements**

### **1. Security**

* Secure password handling and memory sanitization.
* Rate limiting and account lockout after repeated failures.

### **2. Performance**

* Authentication latency must remain **under 300ms under load**.
* Should support **1,000+ concurrent users**.
* Efficient **LDAP connection pooling** and thread management.

### **3. Scalability**

* Stateless interfaces to support horizontal scaling.
* Stateless OTP verification logic (e.g., backed by shared cache like Redis).

### **4. Resilience & Fault Tolerance**

* Handle LDAP outages with retry and **circuit breaker** logic.
* Automatic **failover** between LDAP servers and email providers.

### **5. Maintainability**

* Modular code design to support future enhancements.
* High test coverage with CI/CD (include static code analysis and security scanning).