# Privacy-Preserving Machine Learning Project: Biometric Identification with Homomorphic Encryption

## Introduction

This project focuses on implementing a privacy-preserving biometric identification system using Fully Homomorphic Encryption (FHE). The aim is to secure biometric data—such as fingerprints, iris, or face recognition templates—by conducting computations on encrypted data, preserving privacy and ensuring data security. The project is divided into three parts: limited precision testing on cleartext data, implementing a similarity metric computation over encrypted vectors, and integrating both parts for a full privacy-preserving biometric identification system.

## Subjects and Skills to Learn

1. 1. Coding Languages and Libraries:

- Primary Language: Python (recommended for compatibility with homomorphic encryption libraries).  
- Libraries for Fully Homomorphic Encryption (FHE):  
 - pyhelayers  
 - HEaaN  
 - OpenFHE  
 - Microsoft SEAL (optional, though not actively supported)

1. 2. Concepts:

- Fully Homomorphic Encryption (FHE) with a focus on CKKS scheme.  
- Biometric Identification Algorithms and Vector Embedding.  
- Privacy-Preserving Machine Learning (PPML) fundamentals.

1. 3. Report and Documentation Formats:

- Scientific report written in LaTex (required formats: .pdf, .tex, .bib).  
- README file with setup and execution instructions.  
- Presentation (PPTX) summarizing the project’s background, goals, and results.

## One-Month Accelerated Schedule

### Week 1: Project Setup and Part A  
- \*\*Tasks\*\*:  
 - Familiarize with the biometric dataset, model, and homomorphic encryption.  
 - Implement the biometric identification algorithm with full precision.  
 - Test the model on cleartext data and measure accuracy.  
- \*\*Deliverable\*\*: Short report on initial implementation and results.  
  
### Week 2: Implement Part A with Limited Precision  
- \*\*Tasks\*\*:  
 - Modify the biometric model to operate in a limited precision setting and analyze accuracy.  
- \*\*Deliverable\*\*: Updated report on limited precision results with graphs.  
  
### Week 3: Implement Part B - Homomorphic Similarity Metric  
- \*\*Tasks\*\*:  
 - Install and test FHE libraries.  
 - Implement the homomorphic similarity metric and validate results.  
- \*\*Deliverable\*\*: Report detailing implementation, accuracy, and runtime.  
  
### Week 4: Integrate and Test Part C  
- \*\*Tasks\*\*:  
 - Combine Part A and Part B for the final privacy-preserving biometric identification system.  
 - Test the integrated system, measure accuracy, runtime, and document findings.  
- \*\*Deliverable\*\*: Final project submission with all documentation and files.

## Required Submission Files

- \*\*Scientific Report\*\*: Comprehensive report in .pdf, .tex, .bib formats.  
- \*\*Presentation (PPTX)\*\*: Summarizing background, methodology, results, and conclusions.  
- \*\*Source Code\*\*: Including all Python files, data, and scripts.  
- \*\*README\*\*: Instructions for code execution and setup.  
- \*\*Test Files\*\*: Files to replicate results, including sample data.

## Additional Tips for Success

- \*\*Weekly Meetings\*\*: Consider attending weekly lab meetings to discuss progress.  
- \*\*Backup Work\*\*: Regularly backup code and documents to prevent data loss.  
- \*\*Performance Testing\*\*: Document runtime and accuracy metrics, focusing on both cleartext and encrypted data comparisons.  
- \*\*Ask for Help\*\*: Reach out to TA or lecturer if you encounter technical issues, especially with FHE library configurations.