# Technologies, Languages, and Resources for 'VPN for Office' Project

This document provides a complete list of technologies, programming languages, frameworks, tools, and resources required to learn and build the 'VPN for Office' project. The project will simulate a mini secure VPN system using Java and related technologies, including backend (Spring Boot + Hibernate), frontend (JavaFX / Web), networking, and encryption modules.

## Programming Languages

* • Java SE 17+ (core language)
* • HTML5, CSS3 (for web dashboard if using Thymeleaf or React)
* • JavaScript (for admin frontend if React is used)

## Frameworks & Libraries

* • Spring Boot (for backend REST APIs)
* • Spring Web (for creating REST controllers)
* • Spring Security (for authentication and authorization)
* • Hibernate ORM (for database management)
* • Spring Data JPA (for repository and entity handling)
* • Lombok (to reduce boilerplate code)
* • JavaFX (for desktop client GUI)
* • Jakarta Servlet API (for understanding backend architecture)
* • JWT (JSON Web Token library for secure authentication)
* • BCrypt (for password hashing and encryption)
* • Apache Commons IO / Lang (utility functions)
* • Gson or Jackson (for JSON parsing)

## Networking & Security

* • Java Socket Programming (TCP/UDP communication)
* • SSL/TLS fundamentals
* • AES (Advanced Encryption Standard) encryption
* • RSA encryption for secure key exchange
* • Message Digest (SHA-256, SHA-512)
* • VPN tunneling concepts (virtual private networking principles)
* • Multithreading in networking (ClientHandler, ExecutorService)
* • Firewall and port forwarding concepts (for deployment testing)

## Database & Persistence

* • MySQL or MariaDB (via XAMPP)
* • Hibernate ORM (mapping Java classes to database tables)
* • JPA (Java Persistence API)
* • SQL basics (SELECT, INSERT, UPDATE, DELETE, JOIN)
* • Database design (Entity Relationship Diagram for Users, Sessions, Logs)
* • Flyway or Liquibase (optional for schema migration)

## Frontend & Visualization

* • JavaFX (FXML, SceneBuilder) for desktop UI
* • Thymeleaf (for web-based admin dashboard if using Spring MVC)
* • React.js (optional if using REST + SPA approach)
* • Bootstrap / Tailwind CSS (for styling the admin panel)

## Build, Tools & Configuration

* • Apache Maven (project management and dependency handling)
* • Gradle (optional alternative to Maven)
* • IntelliJ IDEA / Eclipse IDE (Java development)
* • SceneBuilder (for JavaFX UI design)
* • Postman (for API testing)
* • XAMPP (for MySQL + Apache server)
* • Git + GitHub (version control and hosting)
* • Docker (optional, for containerizing the Spring Boot app)
* • JAR packaging and JavaFX bundling (for deployment)

## Deployment & Hosting

* • Render / Railway (free Spring Boot hosting platforms)
* • GitHub Pages / Netlify (if web dashboard is static)
* • JavaFX Runtime packaging (to distribute client app)
* • Cloudflare Tunnel / Ngrok (for testing network connections remotely)

## Learning Resources & References

* • Java Documentation – https://docs.oracle.com/javase/
* • Spring Boot Official Docs – https://spring.io/projects/spring-boot
* • Hibernate ORM Guide – https://hibernate.org/orm/documentation/
* • JavaFX Docs – https://openjfx.io
* • JWT Guide – https://jwt.io/introduction/
* • Baeldung Tutorials – https://www.baeldung.com/
* • GeeksforGeeks Networking in Java – https://www.geeksforgeeks.org/socket-programming-in-java/
* • W3Schools HTML/CSS/JS – https://www.w3schools.com/
* • MySQL Documentation – https://dev.mysql.com/doc/
* • Maven Repository – https://mvnrepository.com/
* • Git & GitHub Guides – https://docs.github.com/en

## Additional Topics to Learn (Advanced Enhancement)

* • Spring Cloud for distributed services (optional future upgrade)
* • WebSocket Communication (for live data updates)
* • Encryption Certificates & KeyStore management (for real SSL)
* • OAuth2 Authentication (alternative to JWT)
* • Logging with Log4j or SLF4J
* • Testing with JUnit & Mockito
* • CI/CD Pipelines using GitHub Actions or Jenkins
* • System Design concepts (Scalability, Reliability, Security)