

VPN For Office Project

Roadmap For next 8 Weeks

Phase 1: Week 1 – Project Setup & Java Networking Basics

Goal: Set up the base project and understand how client–server communication works.

Learn:

- Java Socket Programming (Client/Server Model)
- TCP vs UDP
- Streams & I/O in Java
- Maven Multi-Module Project Structure

Implement:

- Create folder structure: `vpn-server`, `vpn-client`, `vpn-common`
- Build simple server that listens on a port (e.g., `5000`)
- Client connects to server and sends a “HELLO” message
- Server prints “Client connected: [IP]”

Output:

A minimal working client-server connection.



Phase 2: Week 2 – Introduction to Spring Boot & REST APIs

Goal: Learn Spring Boot basics and build a REST API for user login.



Learn:

- Spring Boot basics (controllers, services, repositories)
- REST APIs and JSON
- Spring Boot project structure
- Lombok, application.properties setup



Implement:

- Setup `vpn-server` as a Spring Boot app
- Create endpoints:
 - `/api/test` → returns “Server Running”
- Test using Postman
- Create simple User model + hardcoded list for authentication test



Output:

Spring Boot server that handles simple API calls.

Phase 3: Week 3 – Database & Hibernate Integration

Goal: Connect to MySQL (via XAMPP) and use Hibernate for ORM.

Learn:

- Hibernate Configuration in Spring Boot
- JPA Repository, Entities, Annotations
- Database relationships (OneToMany, etc.)

Implement:

- Connect `vpn-server` to MySQL/MariaDB
- Create `User` table (id, username, password, role)
- Use `JpaRepository` to fetch users
- Test login API with DB credentials

Output:

Database-backed user authentication system.

Phase 4: Week 4 – Authentication & Encryption

Goal: Add secure login with JWT and encrypt communication between client and server.

Learn:

- AES & RSA Encryption in Java
- JWT (JSON Web Token) Authentication
- Password Hashing with BCrypt

Implement:

- Encrypt passwords before saving
- Add JWT authentication to APIs
- Encrypt client-server messages using AES
- Client connects only after successful login

Output:

Secure login and encrypted communication.

Phase 5: Week 5 – Client Application (JavaFX UI)

Goal: Build a desktop UI to simulate the VPN client.

Learn:

- JavaFX basics (FXML, SceneBuilder)
- Event Handling and Background Threads
- HTTP requests in Java

Implement:

- Login screen (username/password)
- “Connect” button to establish VPN session
- “Disconnect” button
- Display status (“Connected”, “Disconnected”)

Output:

Functional desktop client that communicates securely with the backend.

Phase 6: Week 6 – Server Dashboard & Admin Panel

Goal: Create a web admin dashboard to view users, sessions, and logs.

Learn:

- Thymeleaf or React (choose one)
- Fetching data from APIs
- Basic frontend design

Implement:

- `/admin` dashboard for viewing:
 - Active connections
 - Registered users
 - Session logs
- Add “Add User” and “Remove User” options

Output:

Working admin dashboard to manage users and connections.

Phase 7: Week 7 – VPN Tunnel Simulation

Goal: Simulate VPN tunnel behavior (data encryption/decryption + routing visualization).

Learn:

- Simulated network routing in Java
- Packet encryption/decryption workflow
- Multithreading in networking

Implement:

- Simulate sending files/messages through VPN tunnel
- Log encrypted + decrypted data
- Visualize connected clients and data flow in console or UI

Output:

Working VPN tunnel simulation showing encrypted data flow.

Phase 8: Week 8 – Final Touches & Deployment

Goal: Polish, document, and deploy.

Learn:

- Deploying Spring Boot on Render / Railway
- Packaging JavaFX app
- Writing README + documentation







Implement:

- Error handling, validation, cleanup
- Prepare final GitHub repo with README
- Add demo video or screenshots
- Add project to your resume and LinkedIn

Output:

Production-ready mini VPN system for your portfolio.

Final Deliverables

-  **vpn-server**: Spring Boot + Hibernate backend
-  **vpn-client**: JavaFX desktop app
-  **vpn-common**: Shared encryption & utils
-  **vpn-admin**: Web dashboard
-  Full documentation + UML diagrams + database schema
-  Resume-ready project with GitHub + screenshots