### **Đề bài: Xây dựng Dịch vụ IAM với Spring Boot, PostgreSQL và JWT**

**Mục tiêu:**

* Xây dựng một dịch vụ IAM cung cấp các API RESTful cho việc quản lý người dùng và xác thực.
* Áp dụng kiến thức về Spring Boot, Spring Security, JWT, và PostgreSQL.
* Hiểu về các nguyên tắc bảo mật trong phát triển ứng dụng web.

**Yêu cầu chức năng:**

1. **Đăng ký tài khoản (Register):**
   1. Người dùng có thể đăng ký tài khoản mới bằng email và mật khẩu.
   2. Kiểm tra xem email đã tồn tại hay chưa.
   3. Mật khẩu phải được mã hóa (sử dụng BCrypt hoặc Argon2).
2. **Đăng nhập (Login):**
   1. Người dùng có thể đăng nhập bằng email và mật khẩu.
   2. Sau khi đăng nhập thành công, trả về JWT token để sử dụng cho các yêu cầu tiếp theo.
3. **Xác thực bằng JWT:**
   1. Sử dụng JWT để xác thực người dùng cho các API bảo mật.
   2. Token phải có thời hạn và cơ chế làm mới token (refresh token) (nếu có thể).
4. **Xem thông tin cá nhân (Get Profile):**
   1. Người dùng có thể xem thông tin cá nhân của mình.
   2. Bao gồm các thông tin như họ tên, email, số điện thoại, địa chỉ, ngày sinh, ảnh đại diện,...
5. **Cập nhật thông tin cá nhân (Update Profile):**
   1. Người dùng có thể cập nhật thông tin cá nhân của mình.
   2. Kiểm tra và xác thực dữ liệu đầu vào.
6. **Đổi mật khẩu (Change Password):**
   1. Người dùng có thể thay đổi mật khẩu của mình.
   2. Yêu cầu nhập mật khẩu cũ để xác thực.
7. **Quên mật khẩu (Forgot Password):**
   1. Người dùng có thể yêu cầu đặt lại mật khẩu nếu quên.
   2. Gửi email chứa liên kết hoặc mã xác nhận để đặt lại mật khẩu.
   3. Thực hiện đặt lại mật khẩu sau khi xác thực liên kết hoặc mã xác nhận.
8. **Đăng xuất (Logout):**
   1. Thực hiện cơ chế đăng xuất người dùng (nếu có thời gian, triển khai cơ chế hủy token hoặc danh sách đen token).

**Gợi ý thêm các chức năng liên quan tới user và profile:**

* **Xác thực hai yếu tố (Two-Factor Authentication):**
  + Triển khai cơ chế xác thực hai lớp bằng cách gửi mã OTP qua email hoặc SMS.
* **Cập nhật ảnh đại diện (Profile Picture Upload):**
  + Cho phép người dùng tải lên và cập nhật ảnh đại diện.
  + Lưu trữ ảnh trên server hoặc dịch vụ lưu trữ.
* **Thông báo qua email:**
  + Gửi email xác nhận khi đăng ký tài khoản.
  + Gửi email thông báo khi thay đổi thông tin cá nhân hoặc mật khẩu.
* **Hoạt động người dùng (User Activity Logs):**
  + Ghi lại các hoạt động quan trọng của người dùng, như đăng nhập, đăng xuất, thay đổi mật khẩu.

**Yêu cầu kỹ thuật:**

* **Ngôn ngữ và Framework:**
  + Java (khuyến nghị sử dụng Java 17 hoặc mới hơn).
  + Spring Boot (phiên bản mới nhất).
  + Spring Security để quản lý xác thực và phân quyền.
  + Sử dụng JWT cho việc xác thực người dùng.
* **Cơ sở dữ liệu:**
  + PostgreSQL để lưu trữ thông tin người dùng và các dữ liệu liên quan.
  + Sử dụng Spring Data JPA hoặc Hibernate để tương tác với cơ sở dữ liệu.
* **Bảo mật:**
  + Mã hóa mật khẩu người dùng (sử dụng BCryptPasswordEncoder).
  + Bảo vệ các API bằng cách sử dụng cơ chế xác thực và phân quyền của Spring Security.
  + Kiểm tra và ngăn chặn các lỗ hổng bảo mật phổ biến (SQL Injection, XSS, CSRF).
  + Sử dụng HTTPS (SSL/TLS) cho giao tiếp an toàn (nếu có thể).
* **API RESTful:**
  + Tuân thủ các nguyên tắc thiết kế API RESTful.
  + Sử dụng các phương thức HTTP phù hợp (GET, POST, PUT, DELETE).
  + Trả về mã trạng thái HTTP tương ứng với kết quả thực hiện.
* **Xử lý lỗi và thông báo:**
  + Xử lý và trả về thông báo lỗi rõ ràng cho người dùng khi có lỗi xảy ra.
  + Sử dụng cơ chế Exception Handler của Spring Boot.