*Trong ngành IT:*

* ***PM****: Viết tắt của* ***Project Manager*** *(Quản lý Dự án). Người chịu trách nhiệm lập kế hoạch, thực hiện và giám sát dự án, đảm bảo rằng dự án hoàn thành đúng hạn, trong ngân sách và đáp ứng các yêu cầu chất lượng.*
* ***TA****: Viết tắt của* ***Technical Architect*** *(Kiến trúc Sư Kỹ thuật) hoặc* ***Test Analyst*** *(Phân tích Viên Kiểm thử), tùy vào ngữ cảnh. Nếu là* ***Technical Architect****, họ thiết kế kiến trúc tổng thể của hệ thống và đảm bảo rằng các công nghệ được sử dụng là phù hợp và hiệu quả. Nếu là* ***Test Analyst****, họ đảm nhiệm việc thiết kế và thực hiện các kế hoạch kiểm thử để đảm bảo chất lượng phần mềm.*

*https://drive.google.com/drive/folders/1DODDSoNzQWrser\_hYxQiY03WxMvbHdYW?usp=sharing*

*I. Spring security và các thành phần*

* *Spring Security là một framework mạnh mẽ trong Spring Boot, cung cấp các tính năng bảo mật cho ứng dụng web, bao gồm xác thực và phân quyền*

### ***1. Authentication (Xác thực)***

* ***AuthenticationManager****: interface chính để thực hiện xác thực người dùng.*
* ***UserDetailsService****: interface cho việc lấy thông tin người dùng. Bạn có thể triển khai nó để lấy người dùng từ cơ sở dữ liệu.*
* ***UserDetails****: interface cho thông tin người dùng (username, password, roles, etc.).*

### ***2. Authorization (Phân quyền)***

* ***AccessDecisionManager****: Xác định xem người dùng có quyền truy cập vào tài nguyên hay không.*
* ***GrantedAuthority****: Đại diện cho quyền hạn của người dùng (role).*
* ***SecurityExpression****: Các biểu thức để kiểm tra quyền truy cập trong các phương thức hoặc trong JSP/Thymeleaf.*

### ***3. Security Context***

* ***SecurityContext****: Lưu trữ thông tin xác thực của người dùng hiện tại trong một phiên.*
* ***SecurityContextHolder****: Cung cấp quyền truy cập vào SecurityContext cho các thành phần khác trong ứng dụng.*

### ***4. Filter (Bộ lọc)***

* ***OncePerRequestFilter****: Làm việc với các yêu cầu HTTP. Bạn có thể tạo các bộ lọc tùy chỉnh để thực hiện xác thực hoặc phân quyền.*
* ***UsernamePasswordAuthenticationFilter****: Bộ lọc mặc định cho xác thực người dùng bằng tên đăng nhập và mật khẩu.*

### ***5. Session Management (Quản lý phiên)***

* ***SessionRegistry****: Theo dõi thông tin phiên người dùng.*
* ***ConcurrentSessionControlAuthenticationStrategy****: Giới hạn số phiên đồng thời mà một người dùng có thể đăng nhập.*

### ***6. Password Encoding (Mã hóa mật khẩu)***

* ***PasswordEncoder****: Giao diện để mã hóa và kiểm tra mật khẩu. Spring Security cung cấp nhiều triển khai như BCryptPasswordEncoder.*

### ***7. CSRF Protection (Bảo vệ CSRF)***

* ***CsrfFilter****: Bảo vệ ứng dụng khỏi các cuộc tấn công CSRF (Cross-Site Request Forgery).*

### ***8. OAuth2 và JWT***

* ***OAuth2Client****: Hỗ trợ cho xác thực dựa trên OAuth2.*
* ***JwtAuthenticationFilter****: Bộ lọc cho xác thực bằng JSON Web Tokens (JWT).*

### ***9. Method Security (Bảo mật phương thức)***

* *Cung cấp khả năng kiểm tra quyền truy cập vào các phương thức thông qua các chú thích như @PreAuthorize, @PostAuthorize.*

*II. Các giải thích*

1. *PasswordEncoder:*

* *là một interface trong Spring Security, cung cấp các phương thức để mã hóa mật khẩu và kiểm tra mật khẩu đã được mã hóa.*
* *Hai phương thức chính thường được sử dụng là:*
* *encode(CharSequence rawPassword): Mã hóa mật khẩu thô (plain password).*
* *matches(CharSequence rawPassword, String encodedPassword): Kiểm tra xem mật khẩu thô có khớp với mật khẩu đã được mã hóa hay không.*

### ***2. BCryptPasswordEncoder***

* *new BCryptPasswordEncoder(): Đây là một triển khai cụ thể của interface PasswordEncoder.*
* *BCrypt là một thuật toán mã hóa mạnh mẽ, rất được khuyến nghị cho việc bảo mật mật khẩu. Nó có khả năng chống lại các cuộc tấn công brute-force và các phương pháp tấn công khác nhờ vào tính năng tạo ra một "salt" ngẫu nhiên cho mỗi mật khẩu.*
* *Ví dụ minh họa*

*BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();*

*// Mật khẩu cần mã hóa*

*String rawPassword = "mySecretPassword";*

*// Mã hóa mật khẩu*

*String encodedPassword = passwordEncoder.encode(rawPassword);*

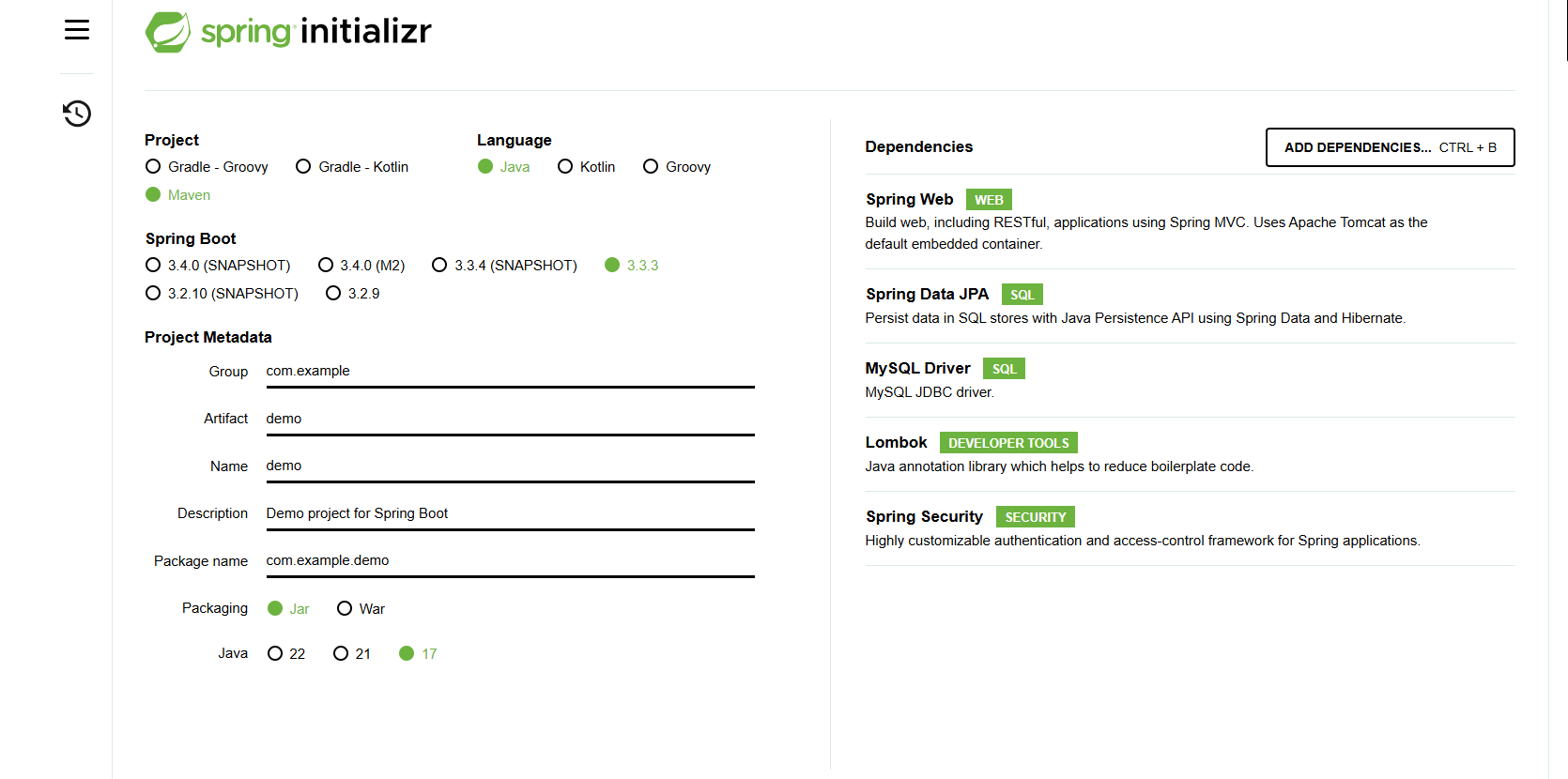
*// Hiển thị chuỗi băm*

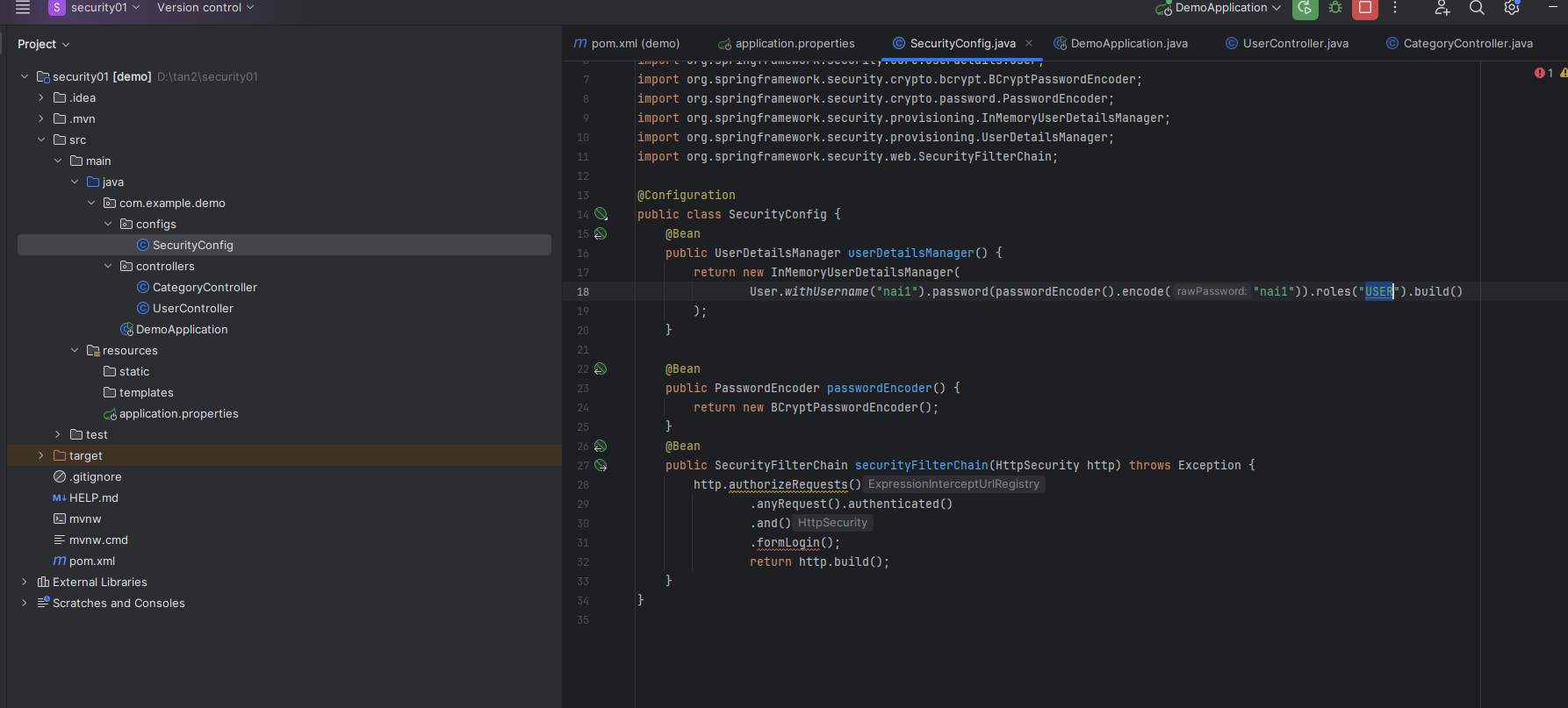
*System.out.println("Encoded Password: " + encodedPassword);*

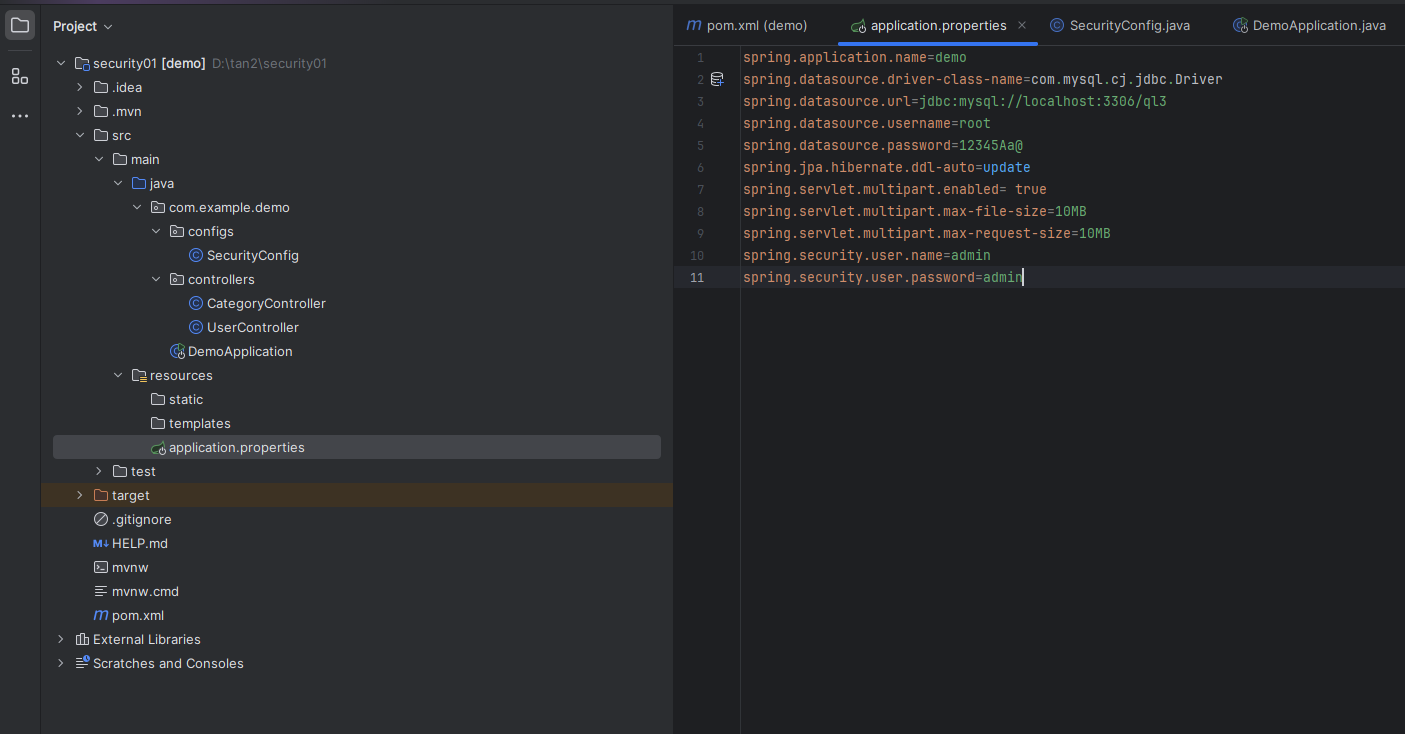
*//Kết quả: Encoded Password: $2a$10$EIXx0z9TpYERl2N8szgrn.w1lScz2a5I53ziTEupBOxC87mtS/1Fe*

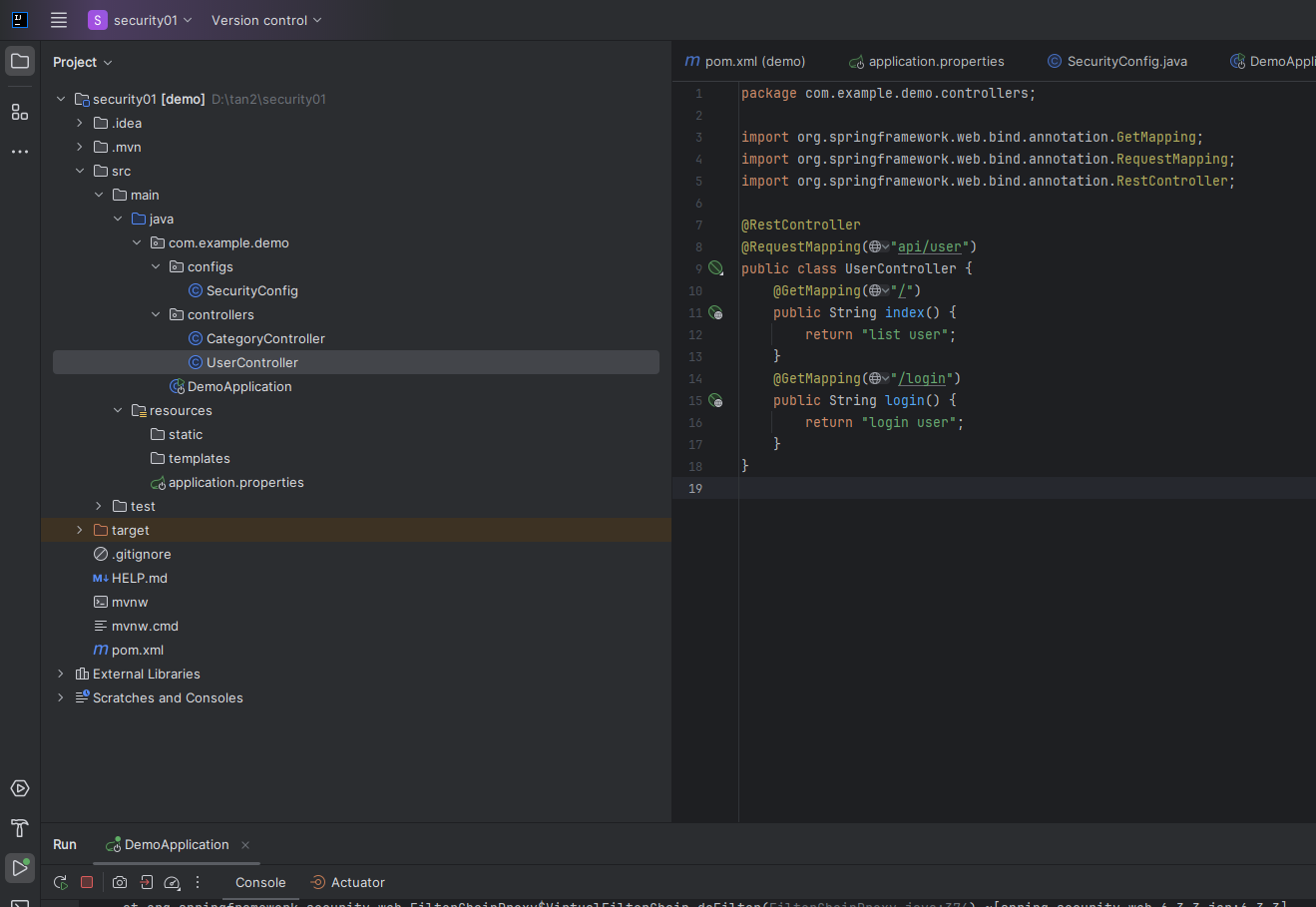
*3. InMemoryUserDetailsManager*

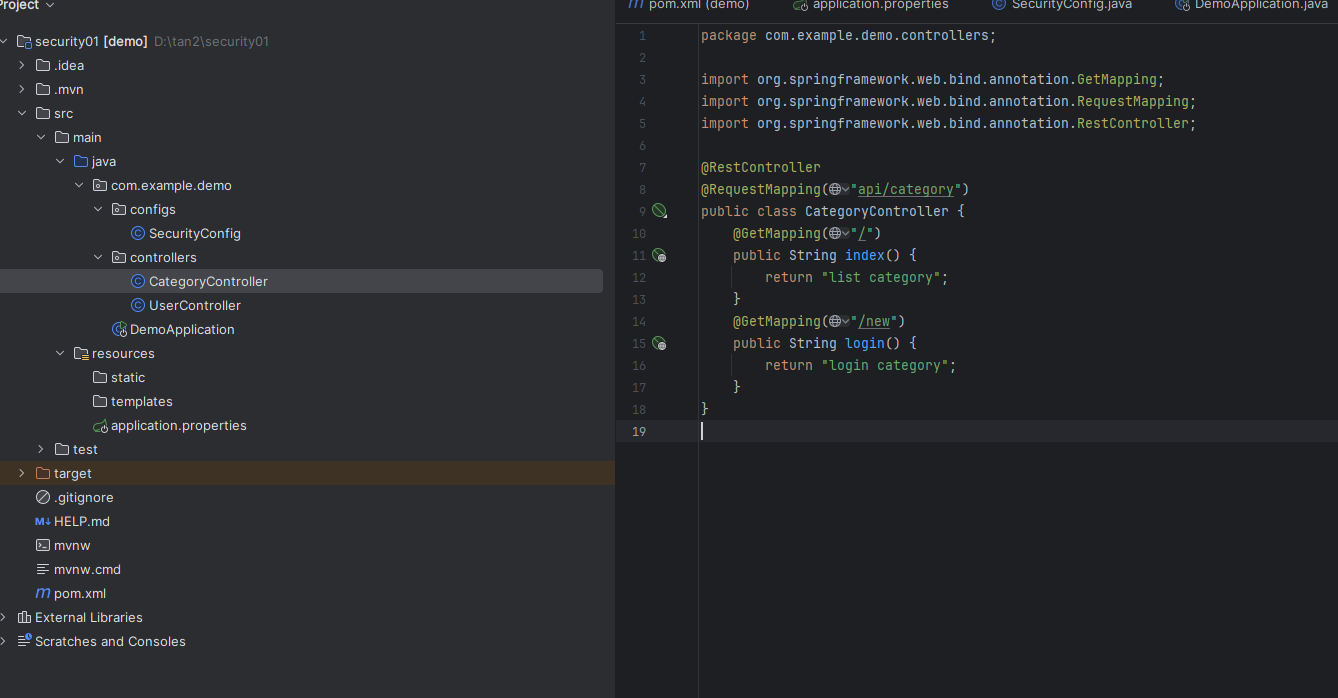
* *InMemoryUserDetailsManager là một lớp trong Spring Security, cung cấp một cách đơn giản để quản lý thông tin người dùng mà không cần phải tương tác với cơ sở dữ liệu.*
* *Thông tin người dùng sẽ được lưu trữ trong bộ nhớ và sẽ không tồn tại khi ứng dụng khởi động lại.*

**

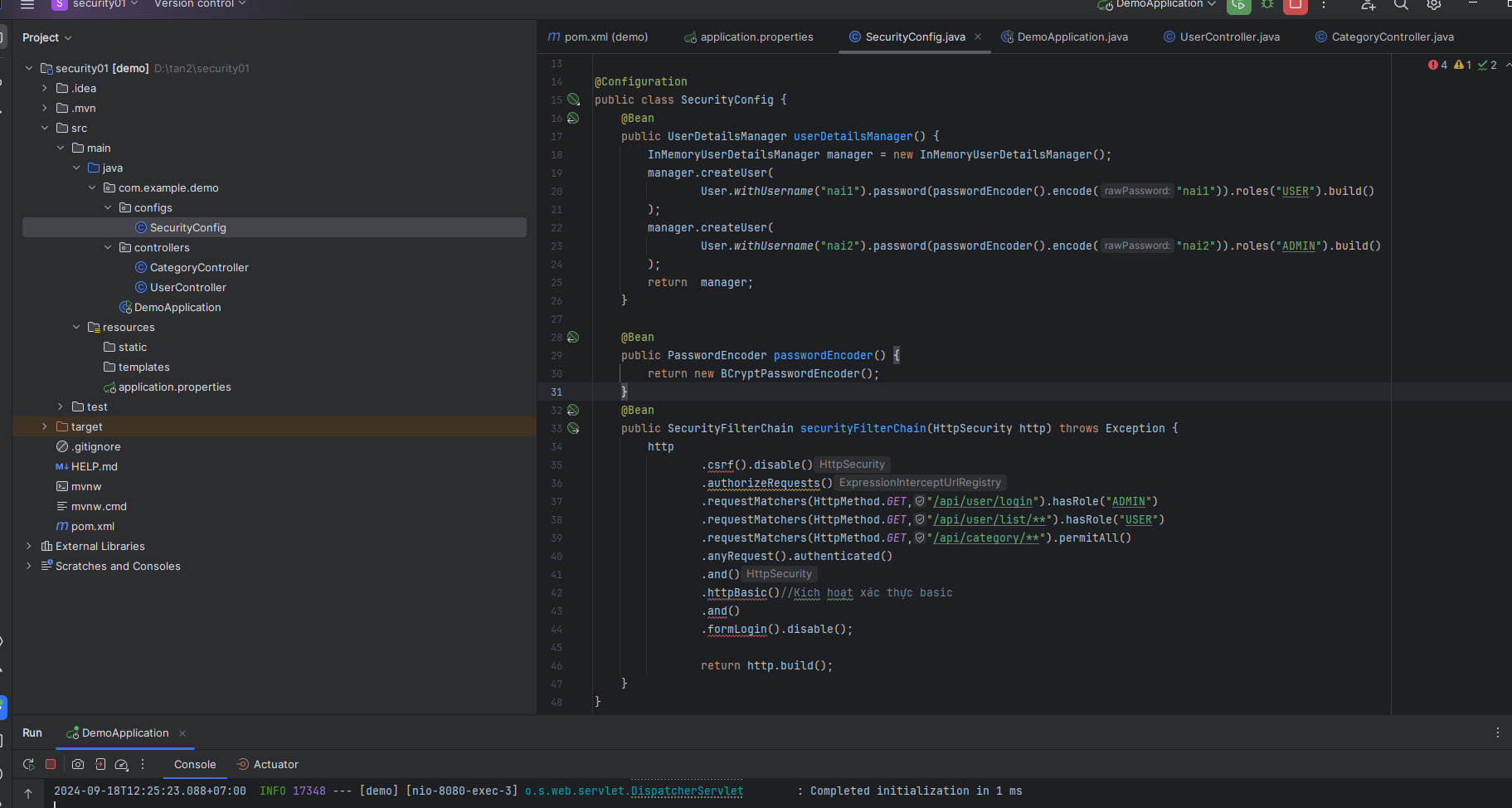
**

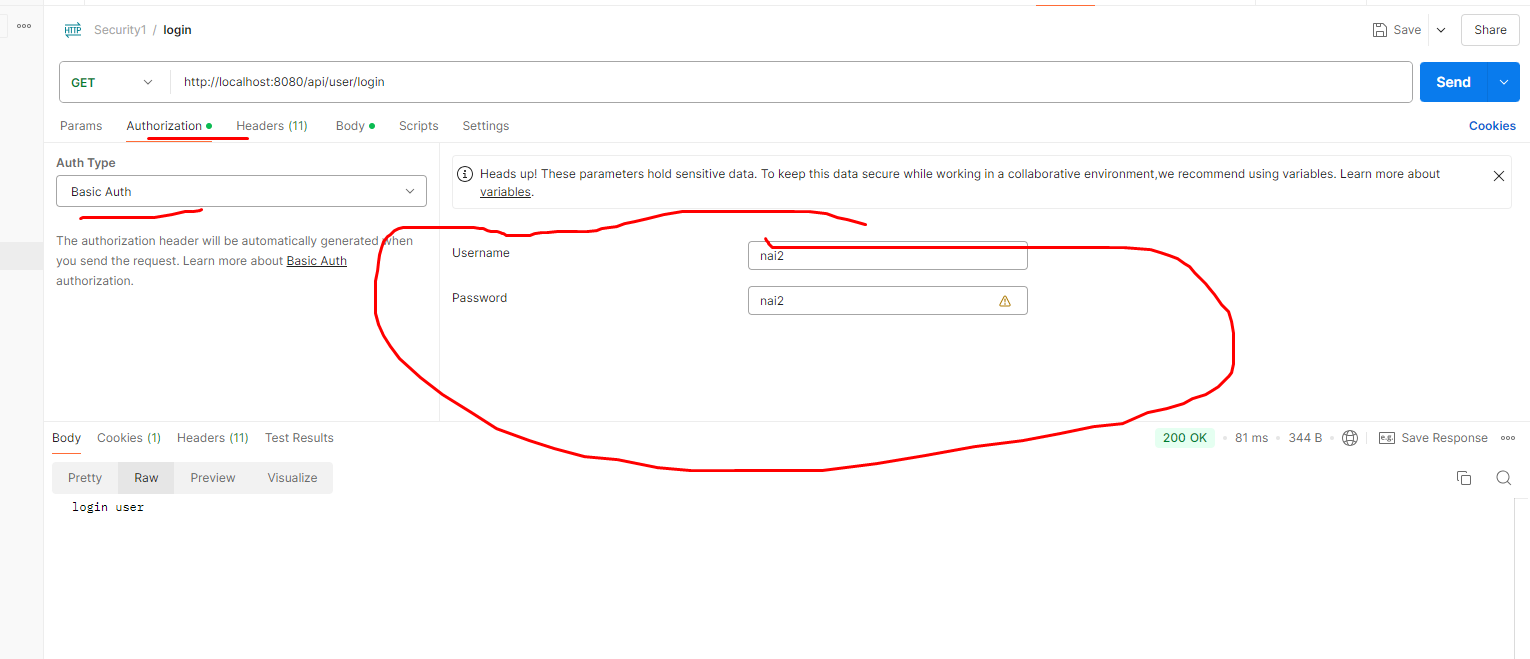
**

**

**

*Vấn đề 2: phần quyền và k dùng form login, mà dùng xác thực authorization basic*

**

**

*Nguyen Hoang Ha*

*Tran Ngoc Trung*

*Tầng 3, Tòa nhà The Mett, 15 Trần Bạch Đằng, Phường Thủ Thiêm, Thành phố Thủ Đức, TP.HCM*

*Nguyen Vinh Quang*

*Nguyen Lam Phuoc Sang*

*Tầng 2, Tòa nhà Centec, 72 - 74 Nguyễn Thị Minh Khai, Phường Võ ThịSáu, Quận 3, TP.HCM*

*Giờ làm việc: sẽ được*

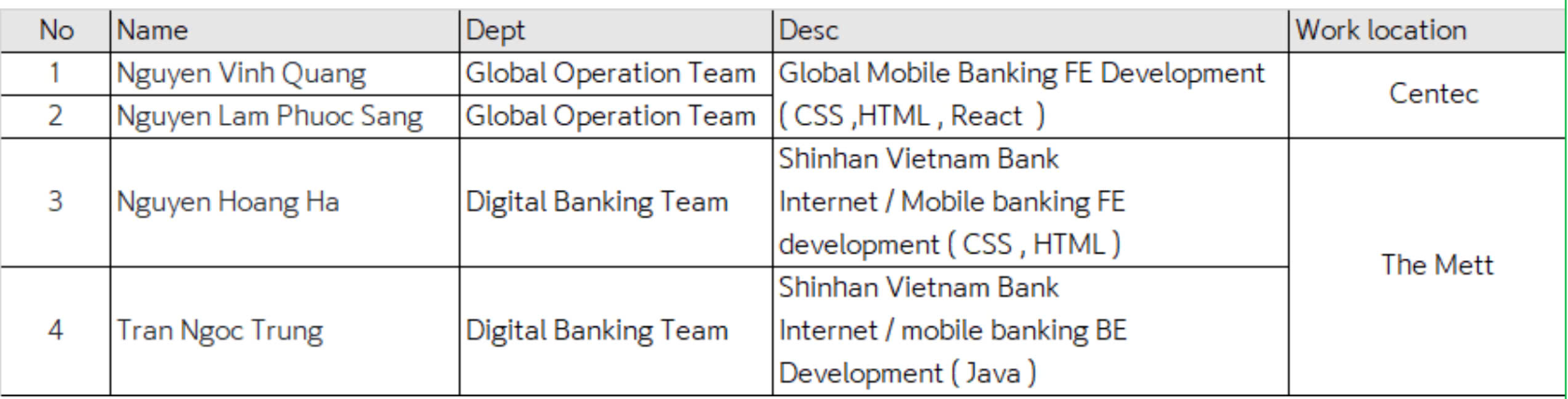
*chia sẻ trong thư mời làm việc.*

*Lương Gross : 5.000.000 VNĐ/tháng*

*Thời gian thực tập: 01/10/2024 đến 31/12/2024 (3 tháng)*

*Thời gian thử việc: 60 ngày (nếu thực tập sinh nào có thành tích tốt và công ty có chỗ trống)*

*Sau khi có thông báo đậu phỏng vấn các bạn sv vẫn phải đi học và điểm danh đầy đủ ạ. Nếu các bạn nghỉ quá số phần trăm đi học vẫn sẽ bị loại không được đi thực tập*

**