International University

School of Computer Science and Engineering

Web Application Development Laboratory

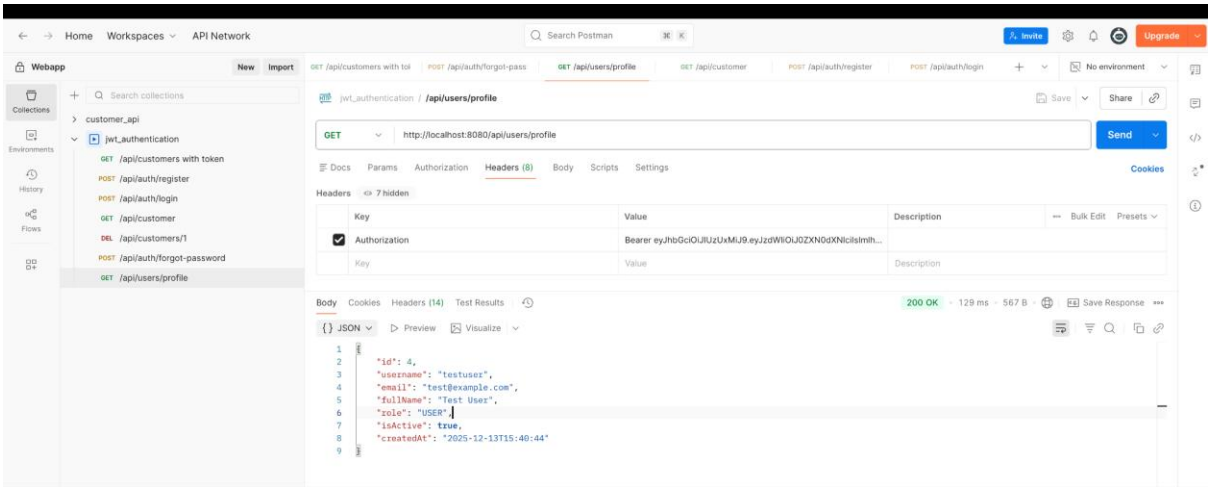
IT093IU

Lab #9

Submitted by

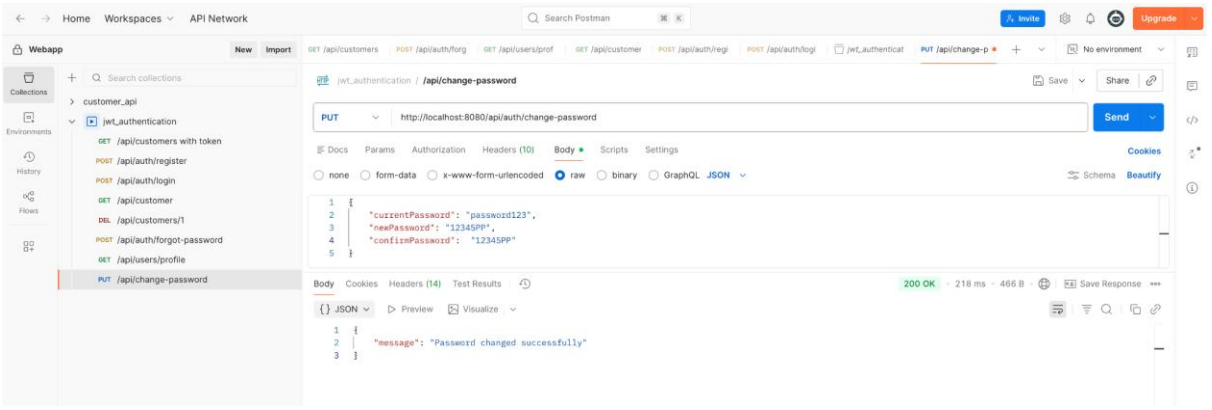
Đặng Nguyễn Nhật Vy - ITITDK22102

Current user profile



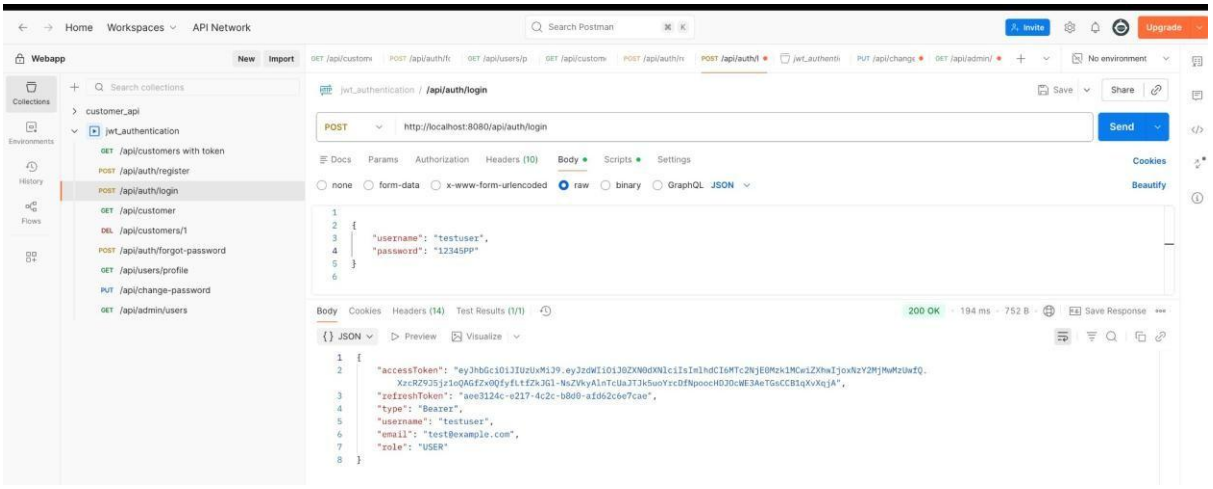
-> This test verifies the functionality of retrieving the currently authenticated user's profile. The request includes a valid JWT token in the Authorization header. Spring Security validates the token, extracts the user identity, and stores it in the SecurityContext. The controller then accesses the authenticated user information and returns a UserResponseDTO containing the user's profile details. A successful 200 OK response confirms that the application correctly identifies the current user based on the JWT token and returns accurate profile data.

Change password success



-> This test validates the change password feature for an authenticated user. The request includes the current password, a new password, and its confirmation. The controller retrieves the current user from the SecurityContext, verifies that the provided current password matches the stored hashed password, and checks that the new password and confirmation are identical. If all validations pass, the new password is hashed using BCrypt and saved to the database. A successful response confirms that the password update process works correctly and securely.

Login with new password



-> This test confirms that the password change has taken effect. The user attempts to log in using the newly updated password. Spring Security authenticates the credentials against the stored hashed password in the database. Since the password was successfully changed in the previous step, authentication succeeds and a new JWT access token is generated and returned. This demonstrates that the password update is persistent and that the user can continue to authenticate normally using the new credentials.