**Лабораторна робота № 5**

##### **Розробка елементів підсистеми захисту програмного продукту від несанкціонованого доступу**

**Мета роботи:** ознайомити студента зі структурою та компонентами підсистеми захисту від несанкціонованого доступу (НСД), яка розробляється як вбудований модуль. Набути практичних навичок з реалізації захисту програмного продукту, який розроблений студентом, від НСД.

**Завдання:** Освоїти та реалізувати на практиці алгоритми захисту власного програмного продукту від НСД.

**1.** **Опис методу реалізації завдання**

Технічний захист від несанкціонованого доступу реалізується за допомогою програмних та апаратних засобів. Система захисту від НСД є комплексом засобів, що призначаються для ускладнення (в ідеальному випадку – для недопущення) нелегального використання (запуску, копіювання) програмного продукту, що захищається.

Пароль – це секретна інформація, якою має володіти тільки авторизований суб’єкт.

Паролем може бути слово, комбінація для замка чи персональний ідентифікаційний номер (PIN).

Парольний механізм може бути досить легко реалізований і мати низьку вартість. Але є суттєві мінуси: зберегти пароль в секреті часто буває проблематично, зловмисники постійно придумують нові методи крадіжки, зламу й підбору пароля. Це обумовлює слабку захищеність парольного механізму.

Для виконання завдання використаємо такі способи підвищення захищеності пароля - Перевірка допустимої кількості разів введення пароля (7 разів).

Розробник повинен мати змогу розблокувати користувача в базі.

**2.** **Алгоритм реалізації завдання**

1. Перед спробою входу користувача перевіримо чи вірно введена комбінація емейлу і пароля
2. Якщо користувач використав неправильні дані то відхиляємо його запит і збільшуємо лічильник неправильних спроб
3. Якщо користувач використав правильні дані то перевіримо чи він вичерпав ліміт введень. Якщо так то відхиляємо його, якщо ні то здійснєюмо його вхід до системи.
4. За проханням користувача адміністратор може скинути кількість спроб

##### **3. Лістинг програми**

Лістинг 1. Модель користувача

**class User < ApplicationRecord**

**MAX\_ATTEMPTS = 7**

**has\_secure\_password**

**validates :name, :email, presence: true**

**validates :email, uniqueness: true, format: { with: URI::MailTo::EMAIL\_REGEXP }**

**validates :password, length: { minimum: 10, maximum: 32 }, if: :password**

**scope :locked, -> { where('attempts >= ?', MAX\_ATTEMPTS) }**

**before\_save { self.email = email.downcase }**

**def locked?**

**attempts >= MAX\_ATTEMPTS**

**end**

**end**

Лістинг 2. Сторінка адміна

**ActiveAdmin.register User do**

**permit\_params :name, :email, :attempts, :password, :password\_confirmation**

**scope I18n.t('activerecord.scopes.user.all'), :all**

**scope I18n.t('activerecord.scopes.user.locked'), :locked**

**filter :name**

**filter :email**

**index do**

**selectable\_column**

**column :id**

**column :name**

**column :email**

**column :locked?**

**actions**

**end**

**form do |f|**

**f.inputs do**

**f.semantic\_errors(\*f.object.errors.keys)**

**f.input :name**

**f.input :email**

**f.input :attempts**

**if f.object.new\_record?**

**f.input :password**

**f.input :password\_confirmation**

**end**

**f.actions**

**end**

**end**

**end**

Лістинг 3. Контролер сесій

**class SessionsController < ApplicationController**

**before\_action :set\_labs, except: :destroy**

**before\_action :set\_user, only: :create**

**def new**

**redirect\_to root\_url if logged\_in?**

**end**

**def create**

**if @user&.authenticate(params[:session][:password])**

**reject\_locked!**

**else**

**@user&.increment!(:attempts) unless @user&.locked?**

**flash.now[:error] = I18n.t('users.sessions.error')**

**render :new**

**end**

**end**

**def destroy**

**log\_out**

**redirect\_to root\_url, notice: I18n.t('users.sessions.logout')**

**end**

**private**

**def set\_user**

**@user = User.find\_by email: params[:session][:email].downcase**

**end**

**def reject\_locked!**

**if @user.locked?**

**redirect\_to new\_session\_path, alert: I18n.t('users.sessions.locked')**

**else**

**log\_in @user**

**redirect\_to root\_url, success: I18n.t('users.sessions.success', name: @user.name)**

**end**

**end**

**end**

Лістинг 4. Допоміжні методи

**module SessionsHelper**

**def log\_in(*user*)**

**session[:user\_id] = user.id**

**end**

**def current\_user**

**@current\_user ||= User.find\_by(id: session[:user\_id]) if session[:user\_id]**

**end**

**def logged\_in?**

**current\_user.present?**

**end**

**def log\_out**

**session.delete(:user\_id)**

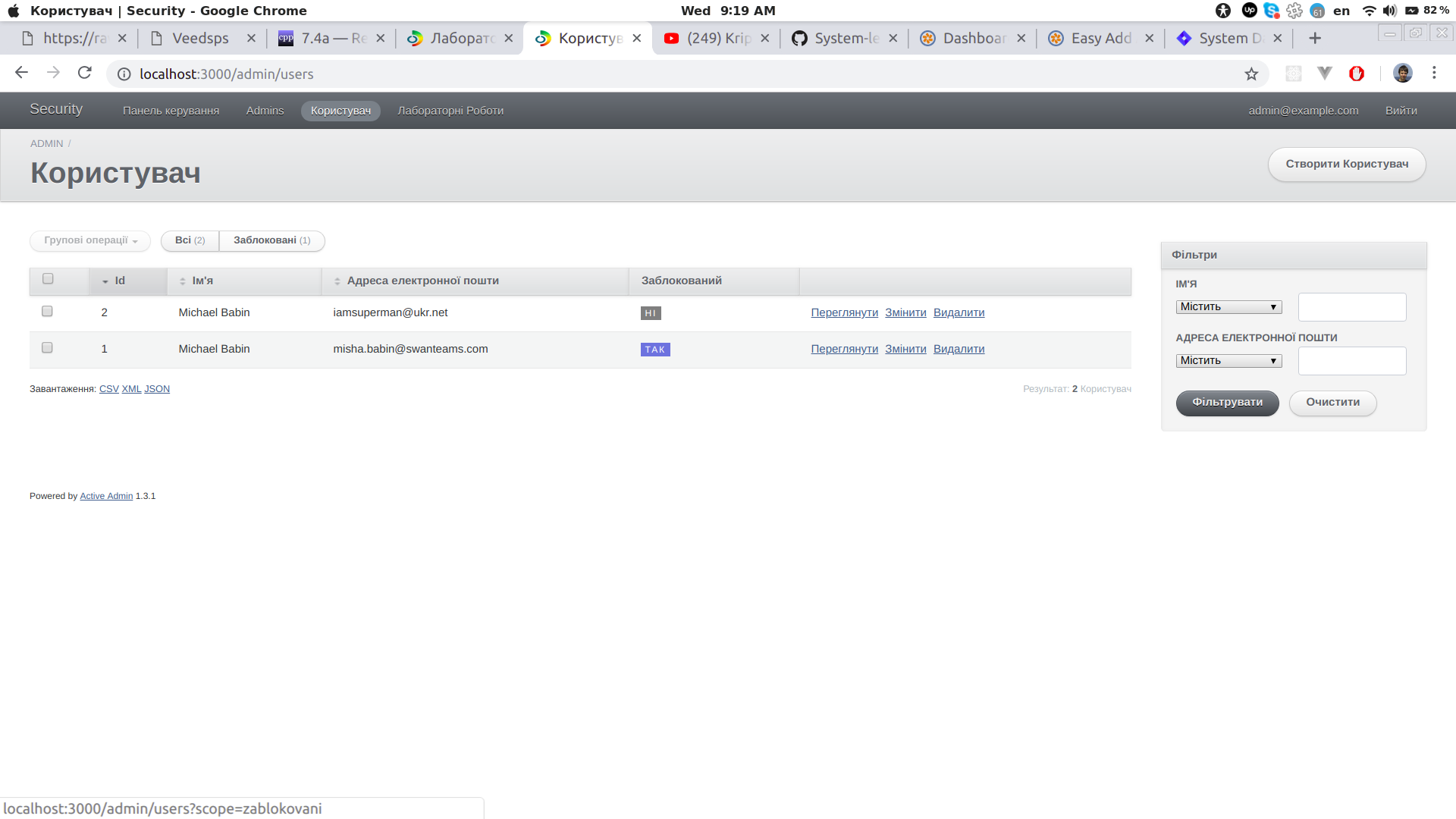
**@current\_user = nil**

**end**

**end**

##### **4. Результати виконання програми**

##### Рис 1. Панель адміністратора

* + - * 1. 

##### Рис 2. Вхід

##### 

##### Рис 3. Акаунт заблокований

##### 

##### Рис 4. Успішний вхід

##### 

##### 

##### **5. Висновок**

В даній лабораторній роботі я освоїв та реалізував на практиці алгоритми захисту власного програмного продукту від НСД методом парольного захисту.

Захист здійснює шифрування паролю, перевірку довжини на перевірку кількості неправильних введень. Розробник програми може власноручно блокувати чи розблокувати користувача.