```
from flask import Blueprint, request, render_template, flash, redirect, url_for, send_file from blueprints.auth.auth_utils import admin_required, login_required, deserialize_user_data from
blueprints.report.report_utils import get_report_by_priority, get_report_by_id, delete_report, get_all_reports, change_report_priority, resolve_report import random, os, pdfkit, socket, shutil
import urllib.request from urllib.parse import urlparse import zipfile from ftplib import FTP from datetime import datetime dashboard_bp = Blueprint('dashboard', __name__,
subdomain='dashboard') pdf_report_path = os.path.join(os.path.dirname(__file__), 'pdf_reports') allowed_hostnames = ['report.comprezzor.htb'] @dashboard_bp.route('/', methods=['GET'])
@admin_required def dashboard(): user_data = request.cookies.get('user_data') user_info = deserialize_user_data(user_data) if user_info['role'] == 'admin': reports = get_report_by_priority(1)
elif user_info['role'] == 'webdev': reports = get_all_reports() return render_template('dashboard/dashboard.html', reports=reports, user_info=user_info) @dashboard_bp.route('/report/',
methods=['GET']) @login_required def get_report(report_id): user_data = request.cookies.get('user_data') user_info = deserialize_user_data(user_data) if user_info['role'] in ['admin', 'webdev']:
report = get_report_by_id(report_id) return render_template('dashboard/report.html', report=report, user_info=user_info) else: pass @dashboard_bp.route('/delete/', methods=['GET'])
@login_required def del_report(report_id): user_data = request.cookies.get('user_data') user_info = deserialize_user_data(user_data) if user_info['role'] in ['admin', 'webdev']: report =
delete_report(report_id) return redirect(url_for('dashboard.dashboard')) else: pass @dashboard_bp.route('/resolve', methods=['POST']) @login_required def resolve(): report_id =
int(request.args.get('report_id')) if resolve_report(report_id): flash('Report resolved successfully!', 'success') else: flash('Error occurred while trying to resolve!', 'error') return
redirect(url_for('dashboard.dashboard')) @dashboard_bp.route('/change_priority', methods=['POST']) @admin_required def change_priority(): user_data = request.cookies.get('user_data')
user_info = deserialize_user_data(user_data) if user_info['role'] != ('webdev' or 'admin'): flash('Not enough permissions. Only admins and webdevs can change report priority.', 'error') return
redirect(url_for('dashboard.dashboard')) report_id = int(request.args.get('report_id')) priority_level = int(request.args.get('priority_level')) if change_report_priority(report_id, priority_level):
flash('Report priority level changed!', 'success') else: flash('Error occurred while trying to change the priority!', 'error') return redirect(url_for('dashboard.dashboard'))
@dashboard_bp.route('/create_pdf_report', methods=['GET', 'POST']) @admin_required def create_pdf_report(): global pdf_report_path if request.method == 'POST': report_url =
request.form.get('report_url') try: scheme = urlparse(report_url).scheme hostname = urlparse(report_url).netloc try: dissallowed_schemas = ["file", "ftp", "ftps"] if (scheme not in
dissallowed_schemas) and ((socket.gethostbyname(hostname.split(":")[0]) != '127.0.0.1') or (hostname in allowed_hostnames)): print(scheme) urllib_request = urllib.request.Request(report_url,
headers={'Cookie':
'user_data=eyJ1c2VyX2lkIjogMSwgInVzZXJuYW1lIjogImFkbWluIiwgInJvbGUiOiAiYWRtaW4ifXwzNDgyMjMzM2Q0NDRhZTBlNDAyMmY2Y2M2NzlhYzlkMjZkMWQxZDY4MmM1OWM2MWNmYm
response = urllib.request.urlopen(urllib_request) html_content = response.read().decode('utf-8') pdf_filename = f'{pdf_report_path}/report_{str(random.randint(10000,90000))}.pdf'
pdfkit.from_string(html_content, pdf_filename) return send_file(pdf_filename, as_attachment=True) except: flash('Unexpected error!', 'error') return
render_template('dashboard/create_pdf_report.html') else: flash('Invalid URL', 'error') return render_template('dashboard/create_pdf_report.html') except Exception as e: raise e else: return
render_template('dashboard/create_pdf_report.html') @dashboard_bp.route('/backup', methods=['GET']) @admin_required def backup(): source_directory =
os.path.abspath(os.path.dirname(__file__) + '../../../') current_datetime = datetime.now().strftime("%Y%m%d%H%M%S") backup_filename = f'app_backup_{current_datetime}.zip' with
zipfile.ZipFile(backup_filename, 'w', zipfile.ZIP_DEFLATED) as zipf: for root, _, files in os.walk(source_directory): for file in files: file_path = os.path.join(root, file) arcname =
os.path.relpath(file_path, source_directory) zipf.write(file_path, arcname=arcname) try: ftp = FTP('ftp.local') ftp.login(user='ftp_admin', passwd='u3jai8y71s2') ftp.cwd('/') with
open(backup_filename, 'rb') as file: ftp.storbinary(f'STOR {backup_filename}', file) ftp.quit() os.remove(backup_filename) flash('Backup and upload completed successfully!', 'success') except
Exception as e: flash(f'Error: {str(e)}', 'error') return redirect(url_for('dashboard.dashboard'))
```