**📄 AccChiTietThanhToanTheoDon.cs (CS, 43 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccChiTietThanhToanTheoDon

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaThanhToan { get; set; }

11 │

12 │ public int? MaDonHang { get; set; }

13 │

14 │ public int? MaHoaDon { get; set; }

15 │

16 │ public decimal? PhaiThu { get; set; }

17 │

18 │ public decimal? DaThanhToan { get; set; }

19 │

20 │ public decimal? ConLai { get; set; }

21 │

22 │ public int? MaTienTeTt { get; set; }

23 │

24 │ public decimal? TyGiaTt { get; set; }

25 │

26 │ public int? UserCreated { get; set; }

27 │

28 │ public DateTime? CreatedDate { get; set; }

29 │

30 │ public int? UserModified { get; set; }

31 │

32 │ public DateTime? ModifiedDate { get; set; }

33 │

34 │ public bool? Deleted { get; set; }

35 │

36 │ public int? UserDeleted { get; set; }

37 │

38 │ public DateTime? DeletedDate { get; set; }

39 │

40 │ public bool? TuDong { get; set; }

41 │

42 │ public decimal? ChenhLechTyGia { get; set; }

43 │ }

────────────────────────────────────────────────────────────

**📄 AccChuyenHinhThucTamUng.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccChuyenHinhThucTamUng

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaChiNhanh { get; set; }

11 │

12 │ public int? MaDatCoc { get; set; }

13 │

14 │ public DateTime? NgayChuyen { get; set; }

15 │

16 │ public string? LyDo { get; set; }

17 │

18 │ public int? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public int? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public int? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 AccCongNoDauKy.cs (CS, 59 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccCongNoDauKy

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaKh { get; set; }

11 │

12 │ public string? TenKh { get; set; }

13 │

14 │ public int? MaDonHang { get; set; }

15 │

16 │ public string? SoDonHang { get; set; }

17 │

18 │ public int? MaHoaDon { get; set; }

19 │

20 │ public string? SoHoaDon { get; set; }

21 │

22 │ public DateTime? NgayHoaDon { get; set; }

23 │

24 │ public DateTime? NgayHetHanTt { get; set; }

25 │

26 │ public int? MaTienTe { get; set; }

27 │

28 │ public int? TyGiaHoaDon { get; set; }

29 │

30 │ public decimal? SoDuNo { get; set; }

31 │

32 │ public int? TyGiaTt { get; set; }

33 │

34 │ public decimal? DaThanhToan { get; set; }

35 │

36 │ public DateTime? NgayThanhToan { get; set; }

37 │

38 │ public string? NguoiThanhToan { get; set; }

39 │

40 │ public int? ThoiHanTt { get; set; }

41 │

42 │ public int? SoNgayNo { get; set; }

43 │

44 │ public string? GhiChu { get; set; }

45 │

46 │ public int? UserCreated { get; set; }

47 │

48 │ public DateTime? CreatedDate { get; set; }

49 │

50 │ public int? UserModified { get; set; }

51 │

52 │ public DateTime? ModifiedDate { get; set; }

53 │

54 │ public bool? Deleted { get; set; }

55 │

56 │ public int? UserDeleted { get; set; }

57 │

58 │ public DateTime? DeletedDate { get; set; }

59 │ }

────────────────────────────────────────────────────────────

**📄 AccHinhThucThu.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccHinhThucThu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? TrangThai { get; set; }

15 │

16 │ public int? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public int? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public int? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 AccLoaiPhieuThu.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccLoaiPhieuThu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? PhieuThanhToan { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public int? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public int? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 AccPhatSinhCongNo.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccPhatSinhCongNo

7 │ {

8 │ public long Ma { get; set; }

9 │

10 │ public DateTime? NgayPhatSinh { get; set; }

11 │

12 │ public string? Code { get; set; }

13 │

14 │ public int? MaKh { get; set; }

15 │

16 │ public string? TenKh { get; set; }

17 │

18 │ public decimal? NoDauKy { get; set; }

19 │

20 │ public decimal? CoDauKy { get; set; }

21 │

22 │ public decimal? NoTrongKy { get; set; }

23 │

24 │ public decimal? CoTrongKy { get; set; }

25 │

26 │ public decimal? NoCuoiKy { get; set; }

27 │

28 │ public decimal? CoCuoiKy { get; set; }

29 │

30 │ public int? UserDeleted { get; set; }

31 │

32 │ public DateTime? DeletedDate { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 AccSoDu131.cs (CS, 21 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccSoDu131

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaKh { get; set; }

11 │

12 │ public string? NgayPhatSinh { get; set; }

13 │

14 │ public decimal? NoCuoiKyNt { get; set; }

15 │

16 │ public decimal? CoCuoiKyNt { get; set; }

17 │

18 │ public decimal? NoCuoiKy { get; set; }

19 │

20 │ public decimal? CoCuoiKy { get; set; }

21 │ }

────────────────────────────────────────────────────────────

**📄 AccSoDu131ChietKhau.cs (CS, 21 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccSoDu131ChietKhau

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public DateOnly? Ngay { get; set; }

11 │

12 │ public int? MaKh { get; set; }

13 │

14 │ public decimal? NoDauKyNt { get; set; }

15 │

16 │ public decimal? NoDauKy { get; set; }

17 │

18 │ public decimal? CoDauKyNt { get; set; }

19 │

20 │ public decimal? CoDauKy { get; set; }

21 │ }

────────────────────────────────────────────────────────────

**📄 AccTaiKhoanNh.cs (CS, 37 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccTaiKhoanNh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? TenNganHang { get; set; }

11 │

12 │ public string? DiaChi { get; set; }

13 │

14 │ public string? SoTaiKhoan { get; set; }

15 │

16 │ public int? TienTe { get; set; }

17 │

18 │ public string? TenChuTaiKhoan { get; set; }

19 │

20 │ public string? GhiChu { get; set; }

21 │

22 │ public int? UserCreated { get; set; }

23 │

24 │ public DateTime? CreatedDate { get; set; }

25 │

26 │ public int? UserModified { get; set; }

27 │

28 │ public DateTime? ModifiedDate { get; set; }

29 │

30 │ public bool? Deleted { get; set; }

31 │

32 │ public int? UserDeleted { get; set; }

33 │

34 │ public DateTime? DeletedDate { get; set; }

35 │

36 │ public int? MaChiNhanh { get; set; }

37 │ }

────────────────────────────────────────────────────────────

**📄 AccTamUng.cs (CS, 59 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccTamUng

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaChiNhanh { get; set; }

11 │

12 │ public int? MaLoaiPhieu { get; set; }

13 │

14 │ public string? SoPhieu { get; set; }

15 │

16 │ public int? MaKh { get; set; }

17 │

18 │ public string? NguoiNop { get; set; }

19 │

20 │ public DateTime? NgayChungTu { get; set; }

21 │

22 │ public int? HinhThucThu { get; set; }

23 │

24 │ public string? LyDo { get; set; }

25 │

26 │ public decimal? SoTien { get; set; }

27 │

28 │ public decimal? DaThanhToan { get; set; }

29 │

30 │ public decimal? ConLai { get; set; }

31 │

32 │ public int? TienTe { get; set; }

33 │

34 │ public int? TyGia { get; set; }

35 │

36 │ public int? MaChuyenTamUng { get; set; }

37 │

38 │ public int? MaDatCoc { get; set; }

39 │

40 │ public int? MaLenhXuatKho { get; set; }

41 │

42 │ public int? MaTaiKhoanNh { get; set; }

43 │

44 │ public int? TrangThai { get; set; }

45 │

46 │ public int? UserCreated { get; set; }

47 │

48 │ public DateTime? CreatedDate { get; set; }

49 │

50 │ public int? UserModified { get; set; }

51 │

52 │ public DateTime? ModifiedDate { get; set; }

53 │

54 │ public bool? Deleted { get; set; }

55 │

56 │ public int? UserDeleted { get; set; }

57 │

58 │ public DateTime? DeletedDate { get; set; }

59 │ }

────────────────────────────────────────────────────────────

**📄 AccThanhToanTheoDon.cs (CS, 59 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccThanhToanTheoDon

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaChiNhanh { get; set; }

11 │

12 │ public int? MaLoaiPhieu { get; set; }

13 │

14 │ public int? HinhThucTt { get; set; }

15 │

16 │ public int? MaTamUng { get; set; }

17 │

18 │ public string? SoPhieu { get; set; }

19 │

20 │ public int? MaKh { get; set; }

21 │

22 │ public DateTime? NgayChungTu { get; set; }

23 │

24 │ public string? NguoiNopTien { get; set; }

25 │

26 │ public string? LyDo { get; set; }

27 │

28 │ public int? TienTe { get; set; }

29 │

30 │ public int? TyGia { get; set; }

31 │

32 │ public decimal? SoTienThanhToan { get; set; }

33 │

34 │ public int? TrangThai { get; set; }

35 │

36 │ public DateTime? NgayChuyenSoKt { get; set; }

37 │

38 │ public int? MaTaiKhoanNh { get; set; }

39 │

40 │ public string? TruongSo1 { get; set; }

41 │

42 │ public string? TruongSo2 { get; set; }

43 │

44 │ public int? UserCreated { get; set; }

45 │

46 │ public DateTime? CreatedDate { get; set; }

47 │

48 │ public int? UserModified { get; set; }

49 │

50 │ public DateTime? ModifiedDate { get; set; }

51 │

52 │ public bool? Deleted { get; set; }

53 │

54 │ public int? UserDeleted { get; set; }

55 │

56 │ public DateTime? DeletedDate { get; set; }

57 │

58 │ public int? KieuPhanBo { get; set; }

59 │ }

────────────────────────────────────────────────────────────

**📄 AccTrangThaiTamUng.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccTrangThaiTamUng

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public int? UserCreated { get; set; }

15 │

16 │ public DateTime? CreatedDate { get; set; }

17 │

18 │ public int? UserModified { get; set; }

19 │

20 │ public DateTime? ModifiedDate { get; set; }

21 │

22 │ public bool? Deleted { get; set; }

23 │

24 │ public int? UserDeleted { get; set; }

25 │

26 │ public DateTime? DeletedDate { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 AccXuLyChenhLechTyGia.cs (CS, 41 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class AccXuLyChenhLechTyGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaHoaDon { get; set; }

11 │

12 │ public int? TienTe { get; set; }

13 │

14 │ public int? TyGiaDonHang { get; set; }

15 │

16 │ public int? TyGiaThanhToan { get; set; }

17 │

18 │ public decimal? PhaiTraUsddonHang { get; set; }

19 │

20 │ public decimal? PhaiTraVnddonHang { get; set; }

21 │

22 │ public decimal? ThanhToanUsddonHang { get; set; }

23 │

24 │ public decimal? ThanhToanVnddonHang { get; set; }

25 │

26 │ public decimal? SoTienXuLyChenhLechVnd { get; set; }

27 │

28 │ public int? UserCreated { get; set; }

29 │

30 │ public DateTime? CreatedDate { get; set; }

31 │

32 │ public int? UserModified { get; set; }

33 │

34 │ public DateTime? ModifiedDate { get; set; }

35 │

36 │ public bool? Deleted { get; set; }

37 │

38 │ public int? UserDeleted { get; set; }

39 │

40 │ public DateTime? DeletedDate { get; set; }

41 │ }

────────────────────────────────────────────────────────────

**📄 AccountController.cs (CS, 156 lines)**

1 │ ﻿using Microsoft.AspNetCore.Authentication;

2 │ using Microsoft.AspNetCore.Authentication.Cookies;

3 │ using Microsoft.AspNetCore.Mvc;

4 │ using Microsoft.Extensions.Logging;

5 │ using Microsoft.IdentityModel.Tokens;

6 │ using System.IdentityModel.Tokens.Jwt;

7 │ using System.Security.Claims;

8 │ using System.Text;

9 │ using System.Threading.Tasks;

10 │ using Microsoft.EntityFrameworkCore;

11 │ using QuanLyChatLuong.Models;

12 │ using System.Security.Cryptography;

13 │ using QuanLyChatLuong.Common.Attributes;

14 │

15 │ namespace QuanLyChatLuong.Controllers

16 │ {

17 │ [Route("account")]

18 │ public class AccountController : Controller

19 │ {

20 │ private readonly ILogger<AccountController> \_logger;

21 │ private readonly QuanLyChatLuongDbContext \_context;

22 │

23 │ public AccountController(ILogger<AccountController> logger, QuanLyChatLuongDbContext context)

24 │ {

25 │ \_logger = logger;

26 │ \_context = context;

27 │ }

28 │

29 │ [HttpGet]

30 │ public IActionResult Login()

31 │ {

32 │ var token = Request.Cookies["Token"];

33 │

34 │ if (string.IsNullOrEmpty(token))

35 │ {

36 │ var branches = \_context.tbl\_ChiNhanhs.ToList();

37 │ return View(branches);

38 │ }

39 │

40 │ return RedirectToAction("Index", "Home");

41 │ }

42 │

43 │ [HttpPost]

44 │ public async Task<IActionResult> LoginAPI(string userName, string password, int branch)

45 │ {

46 │ if (string.IsNullOrEmpty(userName))

47 │ {

48 │ return Json(new { success = false, message = "Tên đăng nhập không được để trống" });

49 │ }

50 │

51 │ var existingUser = await \_context.sys\_Users.FirstOrDefaultAsync(x => x.TenDangNhap == userName);

52 │ if (existingUser == null)

53 │ {

54 │ return Json(new { success = false, message = "Tên đăng nhập không tồn tại" });

55 │ }

56 │

57 │ if (string.IsNullOrEmpty(password))

58 │ {

59 │ return Json(new { success = false, message = "Mật khẩu không được để trống" });

60 │ }

61 │

62 │ bool isPasswordValid = VerifyPasswordWithSHA512(password, existingUser.Password);

63 │ if (!isPasswordValid)

64 │ {

65 │ return Json(new { success = false, message = "Mật khẩu không đúng" });

66 │ }

67 │

68 │ // Kiểm tra mã chi nhánh

69 │ //var userBranch = await \_context.sys\_UserRoles

70 │ // .FirstOrDefaultAsync(x => x.MaUser == existingUser.Ma && x.MaChiNhanh == branch);

71 │

72 │ //if (userBranch == null)

73 │ //{

74 │ // return Json(new { success = false, message = "Người dùng không có quyền truy cập vào chi nh...

75 │ //}

76 │

77 │ var claims = new[] {

78 │ new Claim(ClaimTypes.Name, existingUser.HoTen),

79 │ new Claim(ClaimTypes.NameIdentifier, existingUser.Ma.ToString()),

80 │ new Claim("Branch", branch.ToString()) // Thêm chi nhánh vào claim

81 │ };

82 │

83 │ var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("your-very-secure-secret-key1234567"));

84 │ var creds = new SigningCredentials(key, SecurityAlgorithms.HmacSha256);

85 │

86 │ var claimsIdentity = new ClaimsIdentity(claims, CookieAuthenticationDefaults.AuthenticationScheme);

87 │

88 │ await HttpContext.SignInAsync(

89 │ CookieAuthenticationDefaults.AuthenticationScheme,

90 │ new ClaimsPrincipal(claimsIdentity),

91 │ new AuthenticationProperties

92 │ {

93 │ IsPersistent = true,

94 │ ExpiresUtc = DateTimeOffset.UtcNow.AddMinutes(30)

95 │ });

96 │

97 │ var token = new JwtSecurityToken(

98 │ issuer: "your-issuer",

99 │ audience: "your-audience",

100 │ claims: claims,

101 │ expires: DateTime.Now.AddHours(1),

102 │ signingCredentials: creds

103 │ );

104 │

105 │ var tokenHandler = new JwtSecurityTokenHandler();

106 │ var jwtToken = tokenHandler.WriteToken(token);

107 │

108 │ Response.Cookies.Append("AuthToken", jwtToken, new Microsoft.AspNetCore.Http.CookieOptions

109 │ {

110 │ SameSite = Microsoft.AspNetCore.Http.SameSiteMode.Lax,

111 │ Expires = DateTime.Now.AddHours(1),

112 │ Path = "/",

113 │ });

114 │

115 │ return Ok(new

116 │ {

117 │ success = true,

118 │ message = "Đăng nhập thành công",

119 │ token = jwtToken

120 │ });

121 │ }

122 │

123 │

124 │

125 │ [HttpGet]

126 │ public IActionResult AccessDenied()

127 │ {

128 │ return View();

129 │ }

130 │ [TokenUserCheckHTTP]

131 │ [HttpPost("logout")]

132 │ public IActionResult Logout()

133 │ {

134 │ Response.Cookies.Delete("AuthToken");

135 │

136 │ return Ok(new { success = true });

137 │ }

138 │

139 │ private bool VerifyPasswordWithSHA512(string enteredPassword, string storedHash)

140 │ {

141 │ string enteredPasswordHash = HashPasswordWithSHA512(enteredPassword);

142 │

143 │ return enteredPasswordHash == storedHash;

144 │ }

145 │ private string HashPasswordWithSHA512(string password)

146 │ {

147 │ using (SHA512 shaM = SHA512.Create())

148 │ {

149 │ byte[] bytes = Encoding.UTF8.GetBytes(password);

150 │ byte[] hash = shaM.ComputeHash(bytes);

151 │ return BitConverter.ToString(hash).Replace("-", "").ToUpper();

152 │ }

153 │ }

154 │

155 │ }

156 │ }

────────────────────────────────────────────────────────────

**📄 AuthController.cs (CS, 73 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.IdentityModel.Tokens;

3 │ using QuanLyChatLuong.Models;

4 │ using System;

5 │ using System.IdentityModel.Tokens.Jwt;

6 │ using System.Linq;

7 │ using System.Security.Claims;

8 │ using System.Text;

9 │

10 │ namespace QuanLyChatLuong.Controllers

11 │ {

12 │ public class AuthController : Controller

13 │ {

14 │ private readonly string \_jwtSecretKey = "your-very-secure-secret-key1234567";

15 │

16 │ private readonly QuanLyChatLuongDbContext \_context;

17 │

18 │ public AuthController(QuanLyChatLuongDbContext context)

19 │ {

20 │ \_context = context;

21 │ }

22 │

23 │ private ClaimsPrincipal GetUserFromToken(string token)

24 │ {

25 │ var tokenHandler = new JwtSecurityTokenHandler();

26 │ var key = Encoding.UTF8.GetBytes(\_jwtSecretKey);

27 │

28 │ try

29 │ {

30 │ var principal = tokenHandler.ValidateToken(token, new TokenValidationParameters

31 │ {

32 │ ValidateIssuer = true,

33 │ ValidateAudience = true,

34 │ ValidateLifetime = true,

35 │ ValidIssuer = "your-issuer",

36 │ ValidAudience = "your-audience",

37 │ IssuerSigningKey = new SymmetricSecurityKey(key)

38 │ }, out SecurityToken validatedToken);

39 │

40 │ return principal;

41 │ }

42 │ catch (Exception)

43 │ {

44 │ return null;

45 │ }

46 │ }

47 │

48 │ public IActionResult GetUserInfo(string token)

49 │ {

50 │ var userClaims = GetUserFromToken(token);

51 │ if (userClaims == null)

52 │ {

53 │ return Json(new { success = false, message = "Token không hợp lệ" });

54 │ }

55 │

56 │ var userName = userClaims.FindFirst(ClaimTypes.Name)?.Value;

57 │ var userId = userClaims.FindFirst(ClaimTypes.NameIdentifier)?.Value;

58 │

59 │ var user = \_context.sys\_Users.FirstOrDefault(u => u.HoTen == userName);

60 │ if (user == null)

61 │ {

62 │ return Json(new { success = false, message = "Người dùng không tồn tại" });

63 │ }

64 │

65 │ return Json(new

66 │ {

67 │ success = true,

68 │ userName = userName,

69 │

70 │ });

71 │ }

72 │ }

73 │ }

────────────────────────────────────────────────────────────

**📄 BaoBiController.cs (CS, 169 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyChatLuong.Common;

4 │ using QuanLyChatLuong.Models;

5 │ using QuanLyChatLuong.Models.APIModels;

6 │

7 │ namespace QuanLyChatLuong.Controllers

8 │ {

9 │ [Route("baobi")]

10 │ public class BaoBiController : BaseController

11 │ {

12 │ public BaoBiController(QuanLyChatLuongDbContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, mem...

13 │ {

14 │ m\_MenuID = MenuID.baobi;

15 │ }

16 │

17 │ private ResponseModel CheckRequire(string VatLieuBaoBi)

18 │ {

19 │ if (string.IsNullOrEmpty(VatLieuBaoBi))

20 │ {

21 │ return new ResponseModel(QLCLConst.Error, "Chưa nhập vật liệu bao bì");

22 │ }

23 │ return null;

24 │ }

25 │

26 │ [TokenUserCheckHTTP]

27 │ [HttpGet]

28 │ public IActionResult Index()

29 │ {

30 │

31 │ return View();

32 │ }

33 │

34 │ [TokenUserCheckHTTP(Permission.Add)]

35 │ [HttpGet("add")]

36 │ public IActionResult Add()

37 │ {

38 │

39 │ return View("Form");

40 │ }

41 │

42 │ [TokenUserCheckHTTP()]

43 │ [HttpPost("addAPI")]

44 │ public IActionResult AddAPI(int Ma, string Ten, string VatLieuBaoBi, string TrangThai)

45 │ {

46 │ var result = CheckRequire(VatLieuBaoBi);

47 │

48 │ if (result != null)

49 │ {

50 │ return Json(result);

51 │ }

52 │

53 │ if (dbcontext.bt\_BaoBis.Any(x => x.VatLieuBaoBi == VatLieuBaoBi && x.Ma != Ma))

54 │ {

55 │ return Json(new ResponseModel(QLCLConst.Error, "Vật liệu bao bì đã tồn tại"));

56 │ }

57 │

58 │ var item = new bt\_BaoBi

59 │ {

60 │ Ten = Cast.ToInt(Ten),

61 │ VatLieuBaoBi = VatLieuBaoBi,

62 │ TrangThai = TrangThai == "1",

63 │ NguoiTao = objToken!.userid,

64 │ NgayTao = DateTime.Now,

65 │ MaChiNhanh = objToken.branch

66 │ };

67 │

68 │

69 │ dbcontext.bt\_BaoBis.Add(item);

70 │ dbcontext.SaveChanges();

71 │

72 │ return Json(new ResponseModel(QLCLConst.Successfully, "Thêm thành công"));

73 │ }

74 │

75 │ [TokenUserCheckHTTP(Permission.Edit)]

76 │ [HttpGet("edit")]

77 │ public IActionResult Edit(string id)

78 │ {

79 │ var item = dbcontext.bt\_BaoBis.FirstOrDefault(c => c.Ma == Cast.ToInt(id.Decrypt()));

80 │

81 │ if (item == null)

82 │ {

83 │ return NotFound("Bao bi not found");

84 │ }

85 │

86 │ ViewData["item"] = item;

87 │ return View("Form");

88 │ }

89 │

90 │ [TokenUserCheckHTTP()]

91 │ [HttpPost("updateAPI")]

92 │ public IActionResult UpdateAPI(int Ma, string Ten, string VatLieuBaoBi, string TrangThai)

93 │ {

94 │ var item = dbcontext.bt\_BaoBis.FirstOrDefault(x => x.Ma == Ma && !(x.DaXoa ?? false));

95 │ if (item == null)

96 │ {

97 │ return Json(new ResponseModel(QLCLConst.Error, "Không tồn tại"));

98 │ }

99 │ if (item.VatLieuBaoBi != VatLieuBaoBi && dbcontext.bt\_BaoBis.Any(x => x.VatLieuBaoBi == VatLieuBaoBi && x.Ma...

100 │ {

101 │ return Json(new ResponseModel(QLCLConst.Error, "Vật liệu bao bì đã tồn tại"));

102 │ }

103 │

104 │ item.Ten = Cast.ToInt(Ten);

105 │ item.VatLieuBaoBi = VatLieuBaoBi;

106 │ item.TrangThai = TrangThai == "1";

107 │ item.NguoiSua = objToken!.userid;

108 │ item.NgaySua = DateTime.Now;

109 │ item.MaChiNhanh = objToken.branch;

110 │ try

111 │ {

112 │ dbcontext.bt\_BaoBis.Update(item);

113 │ dbcontext.SaveChanges();

114 │ }

115 │ catch (Exception ex)

116 │ {

117 │ return Json(new ResponseModel(QLCLConst.Error, "Cập nhật thất bại: " + ex.Message));

118 │ }

119 │

120 │ return Json(new ResponseModel(QLCLConst.Successfully, "Cập nhật thành công"));

121 │ }

122 │

123 │ [HttpPost("DeleteAPI")]

124 │ [TokenUserCheckHTTP(Permission.Delete)]

125 │ public IActionResult DeleteAPI(string Ma)

126 │ {

127 │ var item = dbcontext.bt\_BaoBis.FirstOrDefault(x => x.Ma == Cast.ToInt(Ma.Decrypt()));

128 │ if (item == null)

129 │ {

130 │ return Json(QLCLConst.ResponseNotExist);

131 │ }

132 │ item.DaXoa = true;

133 │ dbcontext.SaveChanges();

134 │ return Json(QLCLConst.ResponseDeleteSuccessfully);

135 │ }

136 │

137 │ [TokenUserCheckHTTP(Permission.View)]

138 │ [HttpGet("getListBaoBi")]

139 │ public IActionResult getListBaoBi()

140 │ {

141 │ var listData = dbcontext.bt\_BaoBis.Where(x => !(x.DaXoa ?? false) && x.MaChiNhanh == objToken.branch).OrderB...

142 │ .Select(t => new

143 │ {

144 │ Ma = t.Ma,

145 │ eMa = t.Ma.Encrypt(),

146 │ Ten = t.Ten,

147 │ VatLieuBaoBi = t.VatLieuBaoBi,

148 │ TrangThai = t.TrangThai,

149 │

150 │ }).ToList();

151 │ return Json(new ResponseModel(QLCLConst.Successfully, "", listData));

152 │ }

153 │ [TokenUserCheckHTTP(Permission.View)]

154 │ [HttpPost("getListData")]

155 │ public IActionResult getListData()

156 │ {

157 │ var listData = (from bb in dbcontext.bt\_BaoBis.Where(x => !(x.DaXoa ?? false) && x.MaChiNhanh == objToken.b...

158 │ select new

159 │ {

160 │ ma = bb.Ma,

161 │ eMa = bb.Ma.Encrypt(),

162 │ ten = bb.VatLieuBaoBi,

163 │ kichThuoc = bb.Ten

164 │ }

165 │ ).ToList();

166 │ return Json(new ResponseModelSuccess(QLCLConst.Successfully, listData));

167 │ }

168 │ }

169 │ }

────────────────────────────────────────────────────────────

**📄 BaoCaoBotBienTinhController.cs (CS, 543 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using Microsoft.IdentityModel.Tokens;

4 │ using QuanLyChatLuong.Common;

5 │ using QuanLyChatLuong.Models;

6 │ using QuanLyChatLuong.Models.APIModels;

7 │ using QuanLyChatLuong.Models.CRMEntity;

8 │ using QuanLyChatLuong.Models.StoreProcedures;

9 │ using Spire.Xls;

10 │ using System;

11 │ using System.Data;

12 │ using System.Globalization;

13 │ using static System.Net.WebRequestMethods;

14 │ using Utility = QuanLyChatLuong.Common.Utility;

15 │

16 │ namespace QuanLyChatLuong.Controllers

17 │ {

18 │ [Route("catSoLieuBienTinh")]

19 │ public class BaoCaoBotBienTinhController : BaseController

20 │ {

21 │ private readonly IWebHostEnvironment \_hostingEnvironment;

22 │ private CrmContext crmContext { get; }

23 │

24 │ public BaoCaoBotBienTinhController(QuanLyChatLuongDbContext dbcontext, CrmContext crmdbcontext, IWebHostEnvironm...

25 │ {

26 │ m\_MenuID = MenuID.catSoLieuBienTinh;

27 │ \_hostingEnvironment = hostingEnvironment;

28 │ crmContext = crmdbcontext;

29 │ }

30 │ public class BaoCaoForm

31 │ {

32 │ public string? MaChiNhanh { get; set; }

33 │ public string? MaChiNhanh2 { get; set; }

34 │ public string? Ngay { get; set; } = DateTime.Now.ToString();

35 │ public string? TuNgay { get; set; } = DateTime.Now.ToString();

36 │ public string? DenNgay { get; set; } = DateTime.Now.ToString();

37 │ public string? FileType { get; set; }

38 │ public string? BaoCao { get; set; }

39 │ }

40 │ public class ColorGioiHanDto

41 │ {

42 │ public string MinColor { get; set; }

43 │ public string MaxColor { get; set; }

44 │ public string Code { get; set; }

45 │ public int MaCN { get; set; }

46 │ public string Loai { get; set; }

47 │ public decimal GiaTri { get; set; }

48 │ public string Ten { get; set; }

49 │ }

50 │ [HttpGet]

51 │ [TokenUserCheckHTTP]

52 │ public IActionResult Index()

53 │ {

54 │ ViewData["CurrentUser"] = dbcontext.sys\_Users.FirstOrDefault(x => x.Ma == objToken.userid);

55 │ ViewData["CurrentBranch"] = dbcontext.tbl\_ChiNhanhs.FirstOrDefault(x => x.Ma == objToken.branch);

56 │ return View();

57 │ }

58 │ [HttpPost("BaocaoBotBienTinhTheoNgay")]

59 │ [TokenUserCheckHTTP]

60 │ public IActionResult BaocaoBotBienTinhTheoNgay([FromBody] BaoCaoForm model)

61 │ {

62 │

63 │ var data = dbcontext.Set<sp\_BaoCaoSoLieuBienTinhTheoNgayResult>().FromSqlRaw($"spBaoCaoSoLieuBienTinhTuNgayD...

64 │ int current = (int)objToken.branch;

65 │ string codeChiNhanh = dbcontext.tbl\_ChiNhanhs.FirstOrDefault(x => x.Ma == current).Code;

66 │ var listLoThuMua = (from l in crmContext.CloseChotLo.Where(t => !(t.Deleted ?? false) && (t.TrangThai != (in...

67 │ join cn in crmContext.SysBranch.Where(t => t.Code == codeChiNhanh && !(t.Deleted ?? fals...

68 │ orderby l.CreatedDate

69 │ select new

70 │ {

71 │ malo = l.Ma,

72 │ codelo = l.MaLo + "",

73 │ }).ToList();

74 │

75 │ return Json(new ResponseModel(QLCLConst.OK, "", new

76 │ {

77 │ ListData = data,

78 │ ListLo = listLoThuMua

79 │ }));

80 │ }

81 │ [HttpGet("ExportBaocao")]

82 │ [TokenUserCheckHTTP(Permission.NotCheck)]

83 │ public IActionResult ExportBaocaoBotBienTinhTheoNgay(string BaoCao, string Ngay, string TuNgay, string DenNgay, ...

84 │ {

85 │ int DongBatDau = 0;

86 │ DataTable dt = new DataTable();

87 │ List<ExcelValue> ListGiaTriRiengLe = new List<ExcelValue>();

88 │ string fileTemplate = "", contentType = "", fileName = "", cotKT = "T";

89 │ DateTime dtpTuNgay = DateTime.TryParse(TuNgay, out DateTime tempTuNgay) ? tempTuNgay : DateTime.Now;

90 │ DateTime dtpDenNgay = DateTime.TryParse(DenNgay, out DateTime tempDenNgay) ? tempDenNgay : DateTime.Now;

91 │

92 │ int branch = Cast.ToInt(objToken.branch);

93 │ int maBotBT = Cast.ToInt(MaBotBienTinh);

94 │ int maBotBienTinhBTP = Cast.ToInt(MaBotBienTinhBTP);

95 │

96 │ if (MaChiNhanh != null || !MaChiNhanh.IsEmpty())

97 │ {

98 │ branch = Cast.ToInt(MaChiNhanh);

99 │ }

100 │

101 │ switch (BaoCao)

102 │ {

103 │ case "baocaoBotBienTinhTheoNgay":

104 │ var data = (from sl in dbcontext.bt\_SoLieuBienTinhs

105 │ join ctsl in dbcontext.bt\_ChiTietSoLieus on sl.Ma equals ctsl.MaSoLieu

106 │ join cn in dbcontext.tbl\_ChiNhanhs on sl.MaChiNhanh equals cn.Ma

107 │ where (sl.Ngay != null && (sl.Ngay.Value.Date >= dtpTuNgay.Date && sl.Ngay.Value.Date <=...

108 │ && sl.MaChiNhanh == branch

109 │ && !(ctsl.DaXoa ?? false)

110 │ && !(sl.DaXoa ?? false)

111 │ && sl.Ma == maBotBT

112 │ select new

113 │ {

114 │ Ngay = ctsl.NgayTao,

115 │ Gio = ctsl.Gio,

116 │ Phut = ctsl.Phut,

117 │ Whiteness = ctsl.Whiteness,

118 │ Moisture = ctsl.Moisture,

119 │ Fineness = ctsl.Fineness,

120 │ pHValue = ctsl.pHValue,

121 │ Residue = ctsl.Residue,

122 │ ViscosityPeakBU = ctsl.ViscosityPeakBU,

123 │ ViscosityMax950C0MinBU = ctsl.ViscosityMax950C0MinBU,

124 │ ViscosityMax950C10MinBU = ctsl.ViscosityMax950C10MinBU,

125 │ GelatinnizationTemperature = ctsl.GelatinnizationTemperature,

126 │ AcetylGroup = ctsl.AcetylGroup,

127 │ DS = ctsl.DS,

128 │ Conductivity = ctsl.Conductivity,

129 │ Note = ctsl.Note,

130 │ ViscosityEndPointBU = ctsl.ViscosityEndPointBU,

131 │ ForeignMatter = ctsl.ForeignMatter,

132 │ SO2 = ctsl.SO2,

133 │ StarchContent = ctsl.StarchContent,

134 │ Sensory = ctsl.Sensory,

135 │ DonHang = sl.DonHang,

136 │ TenSanPham = sl.TenSanPham,

137 │ MaLo = sl.MaLo

138 │ }

139 │ ).ToList();

140 │ if (!FileType.IsNullOrEmpty())

141 │ {

142 │ dt.Columns.Add("DATE");

143 │ dt.Columns.Add("HOUR", typeof(int));

144 │ dt.Columns.Add("MINUTE", typeof(int));

145 │ dt.Columns.Add("WHITENESS (%) ", typeof(decimal));

146 │ dt.Columns.Add("MOISTURE (%) ", typeof(decimal));

147 │ dt.Columns.Add("FINENESS (%) ", typeof(decimal));

148 │ dt.Columns.Add("PH", typeof(decimal));

149 │ dt.Columns.Add("RESIDUDE (%) ", typeof(decimal));

150 │ dt.Columns.Add("VISCOSITY PEAK", typeof(decimal));

151 │ dt.Columns.Add("VISCOSITY (START OF HOLDING)", typeof(decimal));

152 │ dt.Columns.Add("VISCOSITY ( START OF COOLING)", typeof(decimal));

153 │ dt.Columns.Add("VISCOSITY ( END OF FINAL)", typeof(decimal));

154 │ dt.Columns.Add("Gelatinization Temperature", typeof(decimal));

155 │ dt.Columns.Add("SO2 (mg/kg) ", typeof(decimal));

156 │ dt.Columns.Add("Acetyl Group %", typeof(decimal));

157 │ dt.Columns.Add("DS", typeof(decimal));

158 │ dt.Columns.Add("Conductivity (mS/Cm)", typeof(decimal));

159 │ dt.Columns.Add("STARCH CONTENT (%)", typeof(decimal));

160 │ dt.Columns.Add("FOREIGN MATTER");

161 │ dt.Columns.Add("SENSORY");

162 │ dt.Columns.Add("Note");

163 │

164 │ foreach (var item in data)

165 │ {

166 │ DateTime? datee = item.Ngay;

167 │ datee ??= DateTime.Now;

168 │ string? formattedDate = datee.Value.ToString("dd/MM/yyyy");

169 │

170 │ dt.Rows.Add(new object[] {

171 │ formattedDate,

172 │ Cast.ToInt(item.Gio),

173 │ Cast.ToInt(item.Phut),

174 │ Cast.ToDouble(item.Whiteness) != 0 ? Cast.ToDouble(item.Whiteness) : DBNull.Value,

175 │ Cast.ToDouble(item.Moisture) != 0 ? Cast.ToDouble(item.Moisture) : DBNull.Value,

176 │ Cast.ToDouble(item.Fineness) != 0 ? Cast.ToDouble(item.Fineness) : DBNull.Value,

177 │ Cast.ToDouble(item.pHValue) != 0 ? Cast.ToDouble(item.pHValue) : DBNull.Value,

178 │ Cast.ToDouble(item.Residue) != 0 ? Cast.ToDouble(item.Residue) : DBNull.Value,

179 │ Cast.ToDouble(item.ViscosityPeakBU) != 0 ? Cast.ToDouble(item.ViscosityPeakBU) : DBNull....

180 │ Cast.ToDouble(item.ViscosityMax950C0MinBU) != 0 ? Cast.ToDouble(item.ViscosityMax950C0Mi...

181 │ Cast.ToDouble(item.ViscosityMax950C10MinBU) != 0 ? Cast.ToDouble(item.ViscosityMax950C10...

182 │ Cast.ToDouble(item.ViscosityEndPointBU) != 0 ? Cast.ToDouble(item.ViscosityEndPointBU) :...

183 │ Cast.ToDouble(item.GelatinnizationTemperature) != 0 ? Cast.ToDouble(item.Gelatinnization...

184 │ Cast.ToDouble(item.SO2) != 0 ? Cast.ToDouble(item.SO2) : DBNull.Value,

185 │ Cast.ToDouble(item.AcetylGroup) != 0 ? Cast.ToDouble(item.AcetylGroup) : DBNull.Value,

186 │ Cast.ToDouble(item.DS) != 0 ? Cast.ToDouble(item.DS) : DBNull.Value,

187 │ Cast.ToDouble(item.Conductivity) != 0 ? Cast.ToDouble(item.Conductivity) : DBNull.Value,

188 │ Cast.ToDouble(item.StarchContent) != 0 ? Cast.ToDouble(item.StarchContent) : DBNull.Valu...

189 │ item.ForeignMatter ,

190 │ item.Sensory ,

191 │ item.Note

192 │ });

193 │ }

194 │

195 │ // Prepare file export

196 │ fileName = FileType == "Excel" ? "BotBienTinhThanhPham.xlsx" : "BotBienTinhThanhPham.pdf";

197 │ contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

198 │

199 │ fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BotBie...

200 │ if (!System.IO.File.Exists(fileTemplate))

201 │ {

202 │ return NotFound("File template not found.");

203 │ }

204 │ ExcelFormat format = new ExcelFormat();

205 │ format.NumberFormat = "#,###0.####";

206 │

207 │ ExcelFormat format2 = new ExcelFormat();

208 │ format2.Border = true;

209 │ ListGiaTriRiengLe = new List<ExcelValue>

210 │ {

211 │ new ExcelValue { Range = "A4:D4", Value = string.Format("Từ ngày {0:dd/MM/yyyy} đế...

212 │ new ExcelValue { Range = "A5:D5", Value = data?.Count>0? "Đơn hàng: " + data?.FirstOr...

213 │ new ExcelValue { Range = "A6:D6", Value = data?.Count>0? "Tên sản phẩm: " + data?.F...

214 │ new ExcelValue { Range = "A7:D7", Value = data?.Count>0? "Lot no: " + data?.FirstOrDefau...

215 │ //new ExcelValue { Range = "A11:U11", Format = format2,KhongGopO = true, KhongDienGiaTri...

216 │ //new ExcelValue { Range = "A11:U11", Format = format,KhongGopO = true, KhongDienGiaTri ...

217 │ };

218 │ DongBatDau = 11;

219 │ cotKT = "U";

220 │ Workbook book = new Workbook();

221 │ book.LoadFromFile(fileTemplate);

222 │ }

223 │ break;

224 │ case "baocaoBotBienTinhBTPTheoNgay":

225 │

226 │ var data2 = (from btp in dbcontext.bt\_BanThanhPhams

227 │ join dv in dbcontext.tbl\_BanThanhPhams on btp.Ma equals dv.MaBanThanhPham

228 │ join cn in dbcontext.tbl\_ChiNhanhs on btp.MaChiNhanh equals cn.Ma

229 │ where (btp.Ngay != null && (btp.Ngay.Value.Date >= dtpTuNgay.Date && btp.Ngay.Value.Dat...

230 │ && btp.MaChiNhanh == branch

231 │ && dv.MaBanThanhPham == maBotBienTinhBTP

232 │ && !(btp.Deleted ?? false) && !(dv.DaXoa ?? false)

233 │ select new

234 │ {

235 │ Gio = dv.Gio,

236 │ Phut = dv.Phut,

237 │ Ngay = dv.CreatedDate,

238 │ ChiNhanh = cn.Ten,

239 │ MaLo = btp.MaLoThuMua,

240 │ MaSP = btp.MaSanPham,

241 │ BonDem = btp.BonDem,

242 │ BonPU = btp.BonPhanUng,

243 │ PhSB = btp.PHSB,

244 │ NDHH = btp.GelatinnizationTemperature,

245 │ Dn = btp.BrabenderViscosity,

246 │ Acetyl = btp.PercentAcetyl,

247 │ DS = dv.DS,

248 │ FreeClo = btp.FreeClo,

249 │ SO2 = dv.SO2,

250 │ Note = dv.Note,

251 │ Ph = dv.pH\_RO,

252 │ HCl = dv.HCL,

253 │ NaOH = dv.NaOH,

254 │ RasperChao = dv.RasperChao,

255 │ CamQuanNaHCO3 = dv.CamQuanNaHCO3,

256 │ pHSuaBot = dv.pHSuaBot,

257 │ BaumeSB = dv.BauMeSuaBot,

258 │ ViscosityEndPointBU = dv.ViscosityEndPointBU,

259 │ ViscosityMax950C0MinBU = dv.ViscosityMax950C0MinBU,

260 │ ViscosityMax950C10MinBU = dv.ViscosityMax950C10MinBU,

261 │ ViscosityPeakBU = dv.ViscosityPeakBU,

262 │ GelatinnizationTemperature = dv.GelatinnizationTemperature,

263 │ AcetylGroup = dv.AcetylGroup,

264 │ Clo = dv.Clo,

265 │ MaSanPham = btp.MaSanPham,

266 │ TinhBotBaMi = dv.TinhBotTrongBaMi

267 │ }

268 │ ).ToList();

269 │ if (!FileType.IsNullOrEmpty())

270 │ {

271 │ dt.Columns.Add("NGÀY");

272 │ dt.Columns.Add("GIỜ", typeof(int));

273 │ dt.Columns.Add("PHÚT", typeof(int));

274 │ dt.Columns.Add("pH RO", typeof(decimal));

275 │ dt.Columns.Add("HCl", typeof(decimal));

276 │ dt.Columns.Add("NaOH", typeof(decimal));

277 │ dt.Columns.Add("Cảm Quan NaHCO3");

278 │ dt.Columns.Add("Rasper cháo", typeof(decimal));

279 │ dt.Columns.Add("Tinh bột trong bã mì (%)");

280 │ dt.Columns.Add("Baume Sữa bột");

281 │ dt.Columns.Add("pH Sữa bột");

282 │ dt.Columns.Add("VISCOSITY PEAK", typeof(decimal));

283 │ dt.Columns.Add("VISCOSITY (START OF HOLDING)", typeof(decimal));

284 │ dt.Columns.Add("VISCOSITY (START OF COOLING)", typeof(decimal));

285 │ dt.Columns.Add("VISCOSITY (END OF FINAL)", typeof(decimal));

286 │ dt.Columns.Add("GELATINIZATION TEMPRATURE", typeof(decimal));

287 │ dt.Columns.Add("SO2", typeof(decimal));

288 │ dt.Columns.Add("ACETYL GROUP", typeof(decimal));

289 │ dt.Columns.Add("DS", typeof(decimal));

290 │ dt.Columns.Add("CLO (%)", typeof(decimal));

291 │ dt.Columns.Add("Note");

292 │ foreach (var item in data2)

293 │ {

294 │ DateTime? datee = item.Ngay;

295 │ datee ??= DateTime.Now;

296 │ string? formattedDate = datee.Value.ToString("dd/MM/yyyy");

297 │

298 │ dt.Rows.Add(new object[] {

299 │ formattedDate,

300 │ Cast.ToInt(item.Gio),

301 │ Cast.ToInt(item.Phut),

302 │ Cast.ToDouble(item.Ph) != 0 ? Cast.ToDouble(item.Ph) : DBNull.Value,

303 │ Cast.ToDouble(item.HCl) != 0 ? Cast.ToDouble(item.HCl) : DBNull.Value,

304 │ Cast.ToDouble(item.NaOH) != 0 ? Cast.ToDouble(item.NaOH) : DBNull.Value,

305 │ item.CamQuanNaHCO3 ,

306 │ Cast.ToDouble(item.RasperChao) != 0 ? Cast.ToDouble(item.RasperChao) : DBNull.Value,

307 │ item.TinhBotBaMi,

308 │ item.BaumeSB,

309 │ item.pHSuaBot,

310 │ Cast.ToDouble(item.ViscosityPeakBU) != 0 ? Cast.ToDouble(item.ViscosityPeakBU) : DBNull....

311 │ Cast.ToDouble(item.ViscosityMax950C0MinBU) != 0 ? Cast.ToDouble(item.ViscosityMax950C0Mi...

312 │ Cast.ToDouble(item.ViscosityMax950C10MinBU) != 0 ? Cast.ToDouble(item.ViscosityMax950C10...

313 │ Cast.ToDouble(item.ViscosityEndPointBU) != 0 ? Cast.ToDouble(item.ViscosityEndPointBU) :...

314 │ Cast.ToDouble(item.GelatinnizationTemperature) != 0 ? Cast.ToDouble(item.Gelatinnization...

315 │ Cast.ToDouble(item.SO2) != 0 ? Cast.ToDouble(item.SO2) : DBNull.Value,

316 │ Cast.ToDouble(item.AcetylGroup) != 0 ? Cast.ToDouble(item.AcetylGroup) : DBNull.Value,

317 │ Cast.ToDouble(item.DS) != 0 ? Cast.ToDouble(item.DS) : DBNull.Value,

318 │ Cast.ToDouble(item.Clo) != 0 ? Cast.ToDouble(item.Clo) : DBNull.Value,

319 │ item.Note

320 │ });

321 │

322 │ // Prepare file export

323 │ fileName = FileType == "Excel" ? "BotBienTinhBanThanhPham.xlsx" : "BotBienTinhBanThanhPham.p...

324 │ contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

325 │

326 │ fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "Bo...

327 │ if (!System.IO.File.Exists(fileTemplate))

328 │ {

329 │ return NotFound("File template not found.");

330 │ }

331 │ ExcelFormat format = new ExcelFormat();

332 │ format.NumberFormat = "#,###0.####";

333 │

334 │ ExcelFormat format2 = new ExcelFormat();

335 │ format2.Border = true;

336 │ ListGiaTriRiengLe = new List<ExcelValue>

337 │ {

338 │ new ExcelValue { Range = "A3:D3", Value = string.Format("Từ ngày {0:dd/MM/yyy...

339 │ new ExcelValue { Range = "A4:D4", Value = data2.Count>0? "Nhà máy: " + data2?....

340 │ new ExcelValue { Range = "A5:D5", Value = data2.Count>0? "Tên sản phẩm: " +...

341 │ new ExcelValue { Range = "A6:D6", Value = data2.Count>0? "Bồn phản ứng: " ...

342 │ new ExcelValue { Range = "A7:D7", Value = data2.Count>0? "Bồn đệm: " + data...

343 │ //new ExcelValue { Range = "A13:U13", Format = format2,KhongGopO = true, KhongDi...

344 │ //new ExcelValue { Range = "A13:U13", Format = format,KhongGopO = true, KhongDie...

345 │ };

346 │ DongBatDau = 13;

347 │ cotKT = "U";

348 │ Workbook book = new Workbook();

349 │ book.LoadFromFile(fileTemplate);

350 │ }

351 │ }

352 │ break;

353 │ case "baocaoBotTuNhienTheoNgay":

354 │ var data3 = (from sl in dbcontext.tbl\_SoLieus

355 │ join ctsl in dbcontext.tbl\_ChiTietSoLieus on sl.Ma equals ctsl.MaSoLieu

356 │ join cn in dbcontext.tbl\_ChiNhanhs on sl.MaChiNhanh equals cn.Ma

357 │ where (sl.Ngay != null && (sl.Ngay.Date >= dtpTuNgay.Date && sl.Ngay.Date <= dtpDenNgay...

358 │ && sl.MaChiNhanh == branch

359 │ && !(ctsl.DaXoa ?? false)

360 │ && !(sl.DaXoa ?? false)

361 │ select new

362 │ {

363 │ Ngay = ctsl.NgayTao,

364 │ Gio = ctsl.Gio,

365 │ Phut = ctsl.Phut,

366 │ Whiteness = ctsl.B\_DoTrang,

367 │ Moisture = ctsl.B\_DoAm,

368 │ Fineness = ctsl.B\_DoMin,

369 │ pHValueBot = ctsl.B\_DoPH,

370 │ pHValueWater = ctsl.NS\_DoPH,

371 │ Residue = ctsl.B\_TapChat,

372 │ ViscosityPeakBU = ctsl.B\_DoKeoBU,

373 │ ViscosityPeakCP = ctsl.B\_DoKeoCP,

374 │ GelatinnizationTemperature = ctsl.ND\_HoHoa,

375 │ DryingTemperature = ctsl.ND\_Say,

376 │ SO2 = ctsl.SO2,

377 │ StarchContent = ctsl.HamLuongTB,

378 │ ForeignMatter = ctsl.ForeignMatter,

379 │ Sensory = ctsl.Sensory,

380 │ Note = ctsl.GhiChu,

381 │ }

382 │ )

383 │ .OrderByDescending(x => x.Ngay)

384 │ .ToList();

385 │ if (!FileType.IsNullOrEmpty())

386 │ {

387 │ dt.Columns.Add("DATE");

388 │ dt.Columns.Add("HOUR", typeof(int));

389 │ dt.Columns.Add("MINUTE", typeof(int));

390 │ dt.Columns.Add("WHITENESS (%) ", typeof(decimal));

391 │ dt.Columns.Add("MOISTURE (%) ", typeof(decimal));

392 │ dt.Columns.Add("FINENESS (%) ", typeof(decimal));

393 │ dt.Columns.Add("PH STARCH", typeof(decimal));

394 │ dt.Columns.Add("PH WATER", typeof(decimal));

395 │ dt.Columns.Add("RESIDUDE (%) ", typeof(decimal));

396 │ dt.Columns.Add("VISCOSITY PEAK (BU)", typeof(decimal));

397 │ dt.Columns.Add("VISCOSITY PEAK (CP)", typeof(decimal));

398 │ dt.Columns.Add("Gelatinization Temperature", typeof(decimal));

399 │ dt.Columns.Add("DRYING Temperature", typeof(decimal));

400 │ dt.Columns.Add("SO2 (mg/kg) ", typeof(decimal));

401 │ dt.Columns.Add("STARCH CONTENT (%)", typeof(decimal));

402 │ dt.Columns.Add("FOREIGN MATTER");

403 │ dt.Columns.Add("SENSORY");

404 │ dt.Columns.Add("Note");

405 │

406 │ foreach (var item in data3)

407 │ {

408 │ DateTime? datee = item.Ngay;

409 │ datee ??= DateTime.Now;

410 │ string? formattedDate = datee.Value.ToString("dd/MM/yyyy");

411 │

412 │ dt.Rows.Add(new object[] {

413 │ formattedDate,

414 │ Cast.ToInt(item.Gio),

415 │ Cast.ToInt(item.Phut),

416 │ Cast.ToDouble(item.Whiteness) != 0 ? Cast.ToDouble(item.Whiteness) : DBNull.Value,

417 │ Cast.ToDouble(item.Moisture) != 0 ? Cast.ToDouble(item.Moisture) : DBNull.Value,

418 │ Cast.ToDouble(item.Fineness) != 0 ? Cast.ToDouble(item.Fineness) : DBNull.Value,

419 │ Cast.ToDouble(item.pHValueBot) != 0 ? Cast.ToDouble(item.pHValueBot) : DBNull.Value,

420 │ Cast.ToDouble(item.pHValueWater) != 0 ? Cast.ToDouble(item.pHValueWater) : DBNull.Value,

421 │ Cast.ToDouble(item.Residue) != 0 ? Cast.ToDouble(item.Residue) : DBNull.Value,

422 │ Cast.ToDouble(item.ViscosityPeakBU) != 0 ? Cast.ToDouble(item.ViscosityPeakBU) : DBNull....

423 │ Cast.ToDouble(item.ViscosityPeakCP) != 0 ? Cast.ToDouble(item.ViscosityPeakCP) : DBNull....

424 │ Cast.ToDouble(item.GelatinnizationTemperature) != 0 ? Cast.ToDouble(item.Gelatinnization...

425 │ Cast.ToDouble(item.DryingTemperature) != 0 ? Cast.ToDouble(item.DryingTemperature) : DBN...

426 │ Cast.ToDouble(item.SO2) != 0 ? Cast.ToDouble(item.SO2) : DBNull.Value,

427 │ Cast.ToDouble(item.StarchContent) != 0 ? Cast.ToDouble(item.StarchContent) : DBNull.Valu...

428 │ item.ForeignMatter ,

429 │ item.Sensory ,

430 │ item.Note

431 │ });

432 │ }

433 │

434 │ // Prepare file export

435 │ fileName = FileType == "Excel" ? "BotTuNhienThanhPham.xlsx" : "BotTuNhienThanhPham.pdf";

436 │ contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

437 │

438 │ fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "BotTuN...

439 │ if (!System.IO.File.Exists(fileTemplate))

440 │ {

441 │ return NotFound("File template not found.");

442 │ }

443 │ ExcelFormat format = new ExcelFormat();

444 │ format.NumberFormat = "#,###0.####";

445 │

446 │ ExcelFormat format2 = new ExcelFormat();

447 │ format2.Border = true;

448 │ ListGiaTriRiengLe = new List<ExcelValue>

449 │ {

450 │ new ExcelValue { Range = "A4:D4", Value = string.Format("Từ ngày {0:dd/MM/yyyy} đế...

451 │ //new ExcelValue { Range = "A5:D5", Value = data?.Count>0? "Đơn hàng: " + data?.First...

452 │ //new ExcelValue { Range = "A6:D6", Value = data?.Count>0? "Tên sản phẩm: " + data?...

453 │ //new ExcelValue { Range = "A7:D7", Value = data?.Count>0? "Lot no: " + data?.FirstOrDef...

454 │ //new ExcelValue { Range = "A11:U11", Format = format2,KhongGopO = true, KhongDienGiaTri...

455 │ //new ExcelValue { Range = "A11:U11", Format = format,KhongGopO = true, KhongDienGiaTri ...

456 │ };

457 │ DongBatDau = 8;

458 │ cotKT = "R";

459 │ Workbook book = new Workbook();

460 │ book.LoadFromFile(fileTemplate);

461 │ }

462 │ break;

463 │ }

464 │ Stream x = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLe, dt, true, DongBatDau, "A", ...

465 │ return new FileStreamResult(x, contentType)

466 │ {

467 │ FileDownloadName = fileName

468 │ };

469 │ }

470 │

471 │ [HttpPost("BaocaoBotBienTinhSSTheoNgay")]

472 │ [TokenUserCheckHTTP]

473 │ public IActionResult BaocaoBotBienTinhSSTheoNgay([FromBody] BaoCaoForm model)

474 │ {

475 │ var data = dbcontext.Set<sp\_BaoCaoSoLieuBienTinhSoSanhTheoNgayResult>().FromSqlRaw($"spBaoCaoSoLieuBienTinhS...

476 │

477 │ return Json(new ResponseModel(QLCLConst.OK, "", data));

478 │ }

479 │ [HttpPost("getColorGioiHanSoLieu")]

480 │ [TokenUserCheckHTTP]

481 │ public IActionResult getColorGioiHanSoLieu([FromBody] BaoCaoForm model)

482 │ {

483 │ int branch = model.MaChiNhanh != null ? Cast.ToInt(model.MaChiNhanh) : objToken.branch;

484 │

485 │ var data = (from gh in dbcontext.tbl\_GioiHanSoLieus.AsNoTracking()

486 │ join c in dbcontext.tbl\_ColorGioiHanSoLieus.AsNoTracking()

487 │ on new { gh.Code, gh.MaChiNhanh }

488 │ equals new { Code = c.Code ?? "", MaChiNhanh = c.MaChiNhanh ?? 0 }

489 │

490 │ where !(gh.DaXoa ?? false) && !(c.DaXoa ?? false) && (gh.TrangThai ?? false) && (branch == 0 || ...

491 │ select new ColorGioiHanDto

492 │ {

493 │ MinColor = c.MinColor,

494 │ MaxColor = c.MaxColor,

495 │ Code = gh.Code,

496 │ MaCN = (int)gh.MaChiNhanh,

497 │ Loai = gh.Loai,

498 │ GiaTri = (int)gh.GiaTri,

499 │ Ten = gh.TenNgan,

500 │ }).ToList(); ;

501 │ return Json(new ResponseModel(QLCLConst.OK, "", data));

502 │ }

503 │ [HttpPost("getBranch")]

504 │ [TokenUserCheckHTTP]

505 │ public IActionResult getBranch()

506 │ {

507 │ bool bShowAll = Utility.checkPermissionIDModel(objToken!.userid, (int)MenuID.catChiTietSoLieuBienTinh, (int)...

508 │ List<tbl\_ChiNhanh> sysBranches = new List<tbl\_ChiNhanh>();

509 │ if (bShowAll)

510 │ {

511 │ sysBranches = dbcontext.tbl\_ChiNhanhs.Where(t => !(t.DaXoa ?? false) && (t.TrangThai ?? false)).ToList()...

512 │ }

513 │ else

514 │ {

515 │ sysBranches = dbcontext.tbl\_ChiNhanhs.Where(t => !(t.DaXoa ?? false) && (t.TrangThai ?? false) && t.Ma =...

516 │ }

517 │ return Json(new ResponseModel(QLCLConst.OK, "", sysBranches.Select(i => new

518 │ {

519 │ ma = i.Ma,

520 │ code = i.Code,

521 │ ten = i.Ten,

522 │ diachi = i.DiaChi,

523 │ trangthai = i.TrangThai,

524 │ })));

525 │ }

526 │ [TokenUserCheckHTTP]

527 │ [HttpPost("getLoThuMua")]

528 │ public IActionResult getLoThuMua()

529 │ {

530 │ string codeChiNhanh = dbcontext.tbl\_ChiNhanhs.FirstOrDefault(x => x.Ma == objToken.branch).Code;

531 │ var listLoThuMua = (from l in crmContext.CloseChotLo.Where(t => !(t.Deleted ?? false) && (t.TrangThai != (in...

532 │ join cn in crmContext.SysBranch.Where(t => t.Code == codeChiNhanh && !(t.Deleted ?? fals...

533 │ orderby l.CreatedDate

534 │ select new

535 │ {

536 │ malo = l.Ma,

537 │ codelo = l.MaLo + "",

538 │ }).ToList();

539 │

540 │ return Json(new ResponseModel("OK", "", listLoThuMua));

541 │ }

542 │ }

543 │ }

────────────────────────────────────────────────────────────

**📄 Base.cs (CS, 70 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using System.Linq.Expressions;

3 │

4 │ namespace QuanLyChatLuong.Models

5 │ {

6 │ public class Base<T> : IDisposable where T : class

7 │ {

8 │ protected readonly DbContext \_dbContext;

9 │ public Base(DbContext dbContext)

10 │ {

11 │ \_dbContext = dbContext;

12 │ }

13 │

14 │ public T First(Expression<Func<T, bool>> predicate)

15 │ {

16 │ try

17 │ {

18 │ return \_dbContext.Set<T>().First(predicate);

19 │ }

20 │ catch

21 │ {

22 │ return null;

23 │ }

24 │ }

25 │

26 │ public int Add(T item, bool isE = false)

27 │ {

28 │ try

29 │ {

30 │ \_dbContext.Set<T>().Add(item);

31 │ return \_dbContext.SaveChanges() > 0 ? 1 : 0;

32 │ }

33 │ catch (Exception e)

34 │ {

35 │ return -1;

36 │ }

37 │ }

38 │

39 │ public int Update(T item, bool isE = false, bool v = false)

40 │ {

41 │ try

42 │ {

43 │ \_dbContext.Set<T>().Update(item);

44 │ return \_dbContext.SaveChanges() > 0 ? 1 : 0;

45 │ }

46 │ catch

47 │ {

48 │ return -1;

49 │ }

50 │ }

51 │ public IQueryable<T> Get(Expression<Func<T, bool>> predicate)

52 │ {

53 │ return \_dbContext.Set<T>().Where(predicate);

54 │ }

55 │ public IQueryable<T> Get()

56 │ {

57 │ return \_dbContext.Set<T>().Where(t => true);

58 │ }

59 │

60 │ public bool Any(Expression<Func<T, bool>> predicate)

61 │ {

62 │ return \_dbContext.Set<T>().Any(predicate);

63 │ }

64 │

65 │ public void Dispose()

66 │ {

67 │ \_dbContext.Dispose();

68 │ }

69 │ }

70 │ }

────────────────────────────────────────────────────────────

**📄 BaseController.cs (CS, 57 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyChatLuong.Common;

3 │ using QuanLyChatLuong.Models;

4 │ using QuanLyChatLuong.Models.APIModels;

5 │

6 │

7 │ namespace QuanLyChatLuong.Controllers

8 │ {

9 │ public abstract class BaseController : Controller

10 │ {

11 │ private UserToken \_objToken;

12 │ public UserToken objToken

13 │ {

14 │ get => \_objToken;

15 │ set

16 │ {

17 │ \_objToken = value;

18 │ dbcontext.userId = \_objToken.userid;

19 │ }

20 │ }

21 │ public MenuID m\_MenuID;

22 │ public bool isAPI;

23 │

24 │ public bool isAdmin

25 │ {

26 │ get

27 │ {

28 │ try

29 │ {

30 │ return dbcontext.vQuyens.Any(t => t.MaQuyen == (int)RoleID.Admin && t.MaMenu == (int)m\_MenuID && t.M...

31 │ }

32 │ catch

33 │ {

34 │ }

35 │ return false;

36 │ }

37 │ }

38 │ public QuanLyChatLuongDbContext dbcontext { get; }

39 │ public MemoryCacheManager memoryCache { get; }

40 │

41 │ public BaseController(QuanLyChatLuongDbContext dbcontext, MemoryCacheManager memoryCache)

42 │ {

43 │ this.dbcontext = dbcontext;

44 │ this.memoryCache = memoryCache;

45 │ }

46 │

47 │ protected override void Dispose(bool disposing)

48 │ {

49 │ if (disposing)

50 │ {

51 │ //dbcontext.Dispose();

52 │ GC.Collect();

53 │ }

54 │ base.Dispose(disposing);

55 │ }

56 │ }

57 │ }

────────────────────────────────────────────────────────────

**📄 BaseRedis.cs (CS, 340 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore.Storage;

2 │ using Newtonsoft.Json;

3 │ using NReJSON;

4 │ using StackExchange.Redis;

5 │ using System;

6 │ using System.Collections.Generic;

7 │ using System.Drawing.Drawing2D;

8 │ using System.Linq;

9 │ using IDatabase = StackExchange.Redis.IDatabase;

10 │

11 │ namespace QuanLyChatLuong.Models

12 │ {

13 │ public class Redis : IDisposable

14 │ {

15 │ private static string redisConnection = "172.16.10.49:6379,password=tri260197,ssl=False,abortConnect=False,conne...

16 │

17 │ private static readonly ConnectionMultiplexer redisConnections = ConnectionMultiplexer.Connect(redisConnection);

18 │

19 │ private static IDatabase db = redisConnections.GetDatabase();

20 │

21 │ public void SetString<T>(string key, T objectToCache, TimeSpan? expire) where T : class

22 │ {

23 │ // IDatabase db = redisConnections.GetDatabase();

24 │ db.StringSet(key, JsonConvert.SerializeObject(objectToCache

25 │ , Formatting.Indented

26 │ , new JsonSerializerSettings

27 │ {

28 │ ReferenceLoopHandling = ReferenceLoopHandling.Serialize,

29 │ PreserveReferencesHandling = PreserveReferencesHandling.Objects

30 │ }));

31 │ if (expire != null)

32 │ {

33 │ db.KeyExpire(key, expire);

34 │ }

35 │ }

36 │

37 │ public T GetString<T>(string key) where T : class

38 │ {

39 │ var redisObject = db.StringGet(key);

40 │ if (redisObject.HasValue)

41 │ {

42 │ return JsonConvert.DeserializeObject<T>(redisObject

43 │ , new JsonSerializerSettings

44 │ {

45 │ ReferenceLoopHandling = ReferenceLoopHandling.Serialize,

46 │ PreserveReferencesHandling = PreserveReferencesHandling.Objects

47 │ });

48 │ }

49 │ else

50 │ {

51 │ return (T)null;

52 │ }

53 │ }

54 │

55 │ public List<T> GetArray<T>(string key) where T : class

56 │ {

57 │ // IDatabase db = redisConnections.GetDatabase();

58 │ RedisResult redisResult = db.JsonGet(key);

59 │ if (!redisResult.IsNull)

60 │ {

61 │ return JsonConvert.DeserializeObject<List<T>>(redisResult.ToString());

62 │ }

63 │ return new List<T>();

64 │ }

65 │ public void SetListArray<T>(string key, List<T> obj, TimeSpan? expire = null) where T : class

66 │ {

67 │ // IDatabase db = redisConnections.GetDatabase();

68 │ //RedisResult redisResult = db.JsonGet(key);

69 │ db.JsonSet(key, JsonConvert.SerializeObject(obj));

70 │ if (expire != null)

71 │ {

72 │ db.KeyExpire(key, expire);

73 │ }

74 │ }

75 │

76 │ public void SetArray<T>(string key, T obj, TimeSpan? expire = null, string properties = "") where T : class

77 │ {

78 │ // IDatabase db = redisConnections.GetDatabase();

79 │ RedisResult redisResult = db.JsonGet(key);

80 │ if (redisResult.IsNull)

81 │ {

82 │ List<T> x = new List<T>

83 │ {

84 │ obj

85 │ };

86 │ db.JsonSet(key, JsonConvert.SerializeObject(x));

87 │ if (expire != null)

88 │ {

89 │ db.KeyExpire(key, expire);

90 │ }

91 │ }

92 │ try

93 │ {

94 │ List<T> data = JsonConvert.DeserializeObject<List<T>>(redisResult.ToString());

95 │ if (data != null && data.Count > 0)

96 │ {

97 │ if (properties != "")

98 │ {

99 │ int i = 0;

100 │ var propertiesInfo = typeof(T).GetProperty(properties);

101 │ object propertiesValue = propertiesInfo.GetValue(obj, null);

102 │ if (propertiesValue == null)

103 │ {

104 │ // throw new Exception("Properties not Found");

105 │ return;

106 │ }

107 │ foreach (T t in data)

108 │ {

109 │ if (propertiesInfo.GetValue(t, null).Equals(propertiesValue))

110 │ {

111 │ db.JsonSet(key, JsonConvert.SerializeObject(obj), "$.[" + i + "]");

112 │ return;

113 │ }

114 │ else

115 │ {

116 │ i++;

117 │ }

118 │ }

119 │ }

120 │ db.JsonArrayAppend(key, ".", JsonConvert.SerializeObject(obj));

121 │ }

122 │ else

123 │ {

124 │ List<T> x = new List<T>

125 │ {

126 │ obj

127 │ };

128 │ db.JsonSet(key, JsonConvert.SerializeObject(x));

129 │ if (expire != null)

130 │ {

131 │ db.KeyExpire(key, expire);

132 │ }

558 │ dt.Columns.Add("UNIT");

559 │ dt.Columns.Add("SPECIFICATION");

560 │ dt.Columns.Add("METHOD");

561 │ dt.Columns.Add("RESULTS");

562 │

563 │ foreach (var item in combineData)

564 │ {

565 │ dt.Rows.Add(new object[]

566 │ {

567 │ item.STT,

568 │ item.Code,

569 │ item.DVT,

570 │ item.TieuChuan,

571 │ item.Method,

572 │ item.KetQua

573 │ });

574 │ }

575 │

576 │ string fileName = FileType == "Excel" ? "KiemTraChatLuongCOA.xlsx" : "KiemTraChatLuongCOA.pdf";

577 │ string contentType = FileType == "Excel" ? "application/ms-excel" : "application/pdf";

578 │

579 │ string fileTemplate = Path.Combine(\_hostingEnvironment.ContentRootPath, "wwwroot", "templates", "KiemTraChat...

580 │ if (!System.IO.File.Exists(fileTemplate))

581 │ {

582 │ return NotFound("File template not found.");

583 │ }

584 │

585 │ ExcelFormat format = new ExcelFormat { NumberFormat = "#,###0.####" };

586 │ ExcelFormat format2 = new ExcelFormat { Border = true };

587 │

588 │ var firstItem = combineData.FirstOrDefault();

589 │ var getTenTiengAnh = !string.IsNullOrWhiteSpace(firstItem?.TenTiengAnh)

590 │ ? firstItem.TenTiengAnh

591 │ : firstItem?.TenSanPham;

592 │

593 │ List<ExcelValue> ListGiaTriRiengLe = new List<ExcelValue>

594 │ {

595 │ new ExcelValue { Range = "E5", Value = getTenTiengAnh },

596 │ new ExcelValue { Range = "E6", Value = firstItem?.MaLo },

597 │ new ExcelValue { Range = "E7", Value = firstItem?.SoLuong },

598 │ new ExcelValue { Range = "E8", Value = firstItem?.NgaySanXuat },

599 │ new ExcelValue { Range = "E9", Value = firstItem?.HanSuDung },

600 │ };

601 │

602 │ int startRow = 12;

603 │ Stream fileStream = ExportToExcel.FromTemplate(fileTemplate, FileType, ListGiaTriRiengLe, dt, true, startRow...

604 │

605 │ return new FileStreamResult(fileStream, contentType)

606 │ {

607 │ FileDownloadName = fileName

608 │ };

609 │ }

610 │

611 │ [HttpPost("getkiemtrachatluongtheongay")]

612 │ [TokenUserCheckHTTP]

613 │ public IActionResult GetKiemTraSoLuongTheoNgay([FromBody] RequestBody model)

614 │ {

615 │ int current = objToken.branch;

616 │ if (!DateTime.TryParse(model.TuNgay, out DateTime dtTuNgay) || !DateTime.TryParse(model.DenNgay, out DateTim...

617 │ {

618 │ return Json(new ResponseModel("Error", "Ngày không hợp lệ."));

619 │ }

620 │

621 │ if (dtTuNgay > dtDenNgay)

622 │ {

623 │ return Json(new ResponseModel("Error", "Từ ngày không lớn hơn Đến Ngày."));

624 │ }

625 │

626 │ var dataKTCL = dbcontext.tbl\_KiemTraChatLuongs

627 │ .Where(x => !(x.DaXoa ?? false)

628 │ && x.NgayTao != null

629 │ && x.NgayTao.Value.Date >= dtTuNgay.Date

630 │ && x.NgayTao.Value.Date <= dtDenNgay.Date

631 │ && x.MaCN == current)

632 │ .OrderByDescending(x => x.NgayTao)

633 │ .ToList()

634 │ .Select(t => new

635 │ {

636 │ Ma = t.Ma,

637 │ eMa = t.Ma.Encrypt(),

638 │ TenSanPham = t.TenSanPham,

639 │ SoDonHang = t.SoDonHang,

640 │ NgayGiaoHang = t.NgayGiaoHang,

641 │ SoLuong = t.SoLuong,

642 │ TenKhachHang = t.TenKhachHang,

643 │ NgaySanXuat = t.NgaySanXuat,

644 │ HanSuDung = t.HanSuDung,

645 │ ChatLuongBot = t.ChatLuongBot,

646 │ KetLuan = t.KetLuan,

647 │ GhiChu = t.GhiChu,

648 │ NguoiKiemTra = t.NguoiKiemTra,

649 │ TrangThai = t.TrangThai,

650 │

651 │ })

652 │ .ToList();

653 │

654 │ var dataCT = new List<tbl\_ChiTietKiemTra>();

655 │

656 │ foreach (var item in dataKTCL)

657 │ {

658 │ var chiTiet = dbcontext.tbl\_ChiTietKiemTras

659 │ .Where(x => x.MaKTCL == item.Ma && !(x.DaXoa ?? false))

660 │ .OrderBy(x => x.NgayTao)

661 │ .ThenBy(x => x.STT)

662 │ .ToList();

663 │

664 │ dataCT.AddRange(chiTiet);

665 │ }

666 │

667 │ return Json(new ResponseModel(QLCLConst.OK, "", new

668 │ {

669 │ ListKTCL = dataKTCL,

670 │ ListCT = dataCT,

671 │ }));

672 │ }

673 │

674 │ [HttpPost("getListChiTietTieuChuan")]

675 │ [TokenUserCheckHTTP]

676 │ public IActionResult GetListChiTietTieuChuan([FromBody] ChiTietTieuChuanDTO request)

677 │ {

678 │ int currentBranch = objToken.branch;

679 │

680 │ var hanghoa = crmContext.VHangHoa

681 │ .Where(x => x.HangHoa ?? false)

682 │ .Select(h => new { h.Ma, h.Ten })

683 │ .AsNoTracking()

684 │ .ToList();

685 │

686 │ var listData = (from tc in dbcontext.bt\_TieuChuans.AsNoTracking()

687 │ join cn in dbcontext.tbl\_ChiNhanhs.AsNoTracking() on tc.MaChiNhanh equals cn.Ma

688 │ join cttc in dbcontext.bt\_ChiTietTieuChuans.AsNoTracking() on tc.Ma equals cttc.MaTieuChuan

689 │ join bb in dbcontext.bt\_BaoBis.AsNoTracking() on tc.MaVatLieuBaoBi equals bb.Ma

690 │ where tc.MaChiNhanh == currentBranch

691 │ && bb.Ten == request.KichThuoc.ToInt()

692 │ && tc.TieuChuan == request.TieuChuan

693 │ && bb.VatLieuBaoBi == request.VatLieuBaoBi

694 │ && !(tc.DaXoa ?? false)

695 │ select new

696 │ {

697 │ cttc.No,

698 │ cttc.TestItem,

699 │ cttc.Unit,

700 │ cttc.Specification,

701 │ cttc.Method,

702 │ tc.MaSanPham

703 │ }).ToList();

704 │

705 │ var combineHangHoaTieuChuan = (from hh in hanghoa

706 │ join tc in listData on hh.Ma equals tc.MaSanPham

707 │ where hh.Ten == request.TenSanPham

708 │ select new

709 │ {

710 │ tc.No,

711 │ tc.TestItem,

712 │ tc.Unit,

713 │ tc.Specification,

714 │ tc.Method,

715 │ }).OrderBy(x => x.No)

716 │ .ToList();

717 │

718 │ return Json(new ResponseModel(QLCLConst.OK, "", combineHangHoaTieuChuan));

719 │ }

720 │ }

721 │ }

────────────────────────────────────────────────────────────

**📄 KiemTraKimLoaiController.cs (CS, 44 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using QuanLyChatLuong.Common;

4 │ using QuanLyChatLuong.Models;

5 │ using QuanLyChatLuong.Models.APIModels;

6 │

7 │ namespace QuanLyChatLuong.Controllers

8 │ {

9 │ [Route("opsKiemTraKimLoai")]

10 │ public class KiemTraKimLoaiController : BaseController

11 │ {

12 │ public KiemTraKimLoaiController(QuanLyChatLuongDbContext dbcontext, MemoryCacheManager memoryCache) : base(dbcon...

13 │ {

14 │ m\_MenuID = MenuID.opsKiemTraKimLoai;

15 │ }

16 │

17 │ [TokenUserCheckHTTP]

18 │ [HttpGet]

19 │ public IActionResult Index()

20 │ {

21 │

22 │ return View();

23 │ }

24 │

25 │ [TokenUserCheckHTTP(Permission.View)]

26 │ [HttpGet("getDeviceList")]

27 │ public IActionResult getDeviceList(string TuNgay, string DenNgay)

28 │ {

29 │ var branchCode = objToken!.branch;

30 │ DateTime fromDate = string.IsNullOrWhiteSpace(TuNgay) ? DateTime.Now : DateTime.Parse(TuNgay);

31 │ DateTime toDate = string.IsNullOrWhiteSpace(DenNgay) ? DateTime.Now : DateTime.Parse(DenNgay);

32 │

33 │ string strFromDate = fromDate.ToString("MM/dd/yyyy");

34 │ string strToDate = toDate.ToString("MM/dd/yyyy");

35 │ var listData = dbcontext.Set<sp\_DeviceListResult>()

36 │ .FromSqlRaw("EXEC sp\_SoLieuKiemTraKimLoai\_TuNgay\_DenNgay @MaCN = {0}, @TuNgay = {1}, @DenNgay = {2}",

37 │ branchCode, strFromDate, strToDate)

38 │ .ToList();

39 │

40 │ return Json(new ResponseModel(QLCLConst.OK, "", listData));

41 │ }

42 │

43 │ }

44 │ }

────────────────────────────────────────────────────────────

**📄 LocCheckIn.cs (CS, 35 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class LocCheckIn

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaNhanVien { get; set; }

11 │

12 │ public int? MaKh { get; set; }

13 │

14 │ public decimal? Longitude { get; set; }

15 │

16 │ public decimal? Latitude { get; set; }

17 │

18 │ public string? Address { get; set; }

19 │

20 │ public string? GhiChu { get; set; }

21 │

22 │ public int? UserCreated { get; set; }

23 │

24 │ public DateTime? CreatedDate { get; set; }

25 │

26 │ public int? UserModified { get; set; }

27 │

28 │ public DateTime? ModifiedDate { get; set; }

29 │

30 │ public bool? Deleted { get; set; }

31 │

32 │ public int? UserDeleted { get; set; }

33 │

34 │ public DateTime? DeletedDate { get; set; }

35 │ }

────────────────────────────────────────────────────────────

**📄 LocHuyenThi.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class LocHuyenThi

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Ten { get; set; }

11 │

12 │ public int? MaTinh { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public int? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public int? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public int? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 LocPhuongXa.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class LocPhuongXa

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Ten { get; set; }

11 │

12 │ public int? MaHuyen { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public int? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public int? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public int? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 LocQuocGia.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class LocQuocGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Ten { get; set; }

11 │

12 │ public bool? Status { get; set; }

13 │

14 │ public int? UserCreated { get; set; }

15 │

16 │ public DateTime? CreatedDate { get; set; }

17 │

18 │ public int? UserModified { get; set; }

19 │

20 │ public DateTime? ModifiedDate { get; set; }

21 │

22 │ public bool? Deleted { get; set; }

23 │

24 │ public int? UserDeleted { get; set; }

25 │

26 │ public DateTime? DeletedDate { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 LocTinhThanh.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class LocTinhThanh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Ten { get; set; }

11 │

12 │ public int? MaQuocGia { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public int? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public int? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public int? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 LoginController.cs (CS, 92 lines)**

1 │ ﻿using System.Net;

2 │ using Microsoft.AspNetCore.Mvc;

3 │ using Microsoft.EntityFrameworkCore;

4 │ using NetTools;

5 │ using Newtonsoft.Json;

6 │ using QuanLyChatLuong.Common;

7 │ using QuanLyChatLuong.Models;

8 │ using QuanLyChatLuong.Models.APIModels;

9 │

10 │ namespace QuanLyChatLuong.Controllers

11 │ {

12 │ public class LoginController : BaseController

13 │ {

14 │ public LoginController(QuanLyChatLuongDbContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, mem...

15 │ {

16 │ }

17 │ public IActionResult Index()

18 │ {

19 │ ViewData["ListBranch"] = dbcontext.tbl\_ChiNhanhs.Where(x => !(x.DaXoa ?? false) && (x.TrangThai ?? false)).T...

20 │ return View();

21 │ }

22 │

23 │ [HttpPost]

24 │ public IActionResult Authenticate(string username, string password, string branch)

25 │ {

26 │ int branchId = Convert.ToInt32(branch);

27 │

28 │ var user = dbcontext.sys\_Users.FirstOrDefault(x => x.TenDangNhap == username);

29 │

30 │ if (user == null) return Json(new ResponseModel("Error", "Tài khoản không tồn tại"));

31 │

32 │ if (user.DaXoa ?? false || !(user.TrangThai ?? false)) return Json(new ResponseModel("Error", "Tài khoản ...

33 │

34 │ if (user != null && user.TenDangNhap == username && user.Password == Utility.GetSHA512(password))

35 │ {

36 │ List<sys\_UserRole> ur = dbcontext.sys\_UserRoles.Where(x => x.MaUser == user.Ma

37 │ && (x.TrangThai ?? false) && !(x.DaXoa ?? false) && x.MaChiNhanh == branchId).ToList();

38 │ if (ur.Count > 0)

39 │ {

40 │ DateTime now = DateTime.Now;

41 │ DateTime expiresDate;

42 │ if (now.Month == 12)

43 │ {

44 │ expiresDate = new DateTime(now.Year + 1, 1, 1);

45 │ }

46 │ else

47 │ {

48 │ expiresDate = new DateTime(now.Year, now.Month + 1, 1);

49 │ }

50 │ string token = signToken(new UserToken(user.Ma, branchId, username));

51 │ Response.Cookies.Append("AuthToken", token, new() { Expires = expiresDate });

52 │ return Json(new ResponseModel("OK", "Đăng nhập thành công"));

53 │

54 │ }

55 │ else return Json(new ResponseModel("Error", "Bạn không thể đăng nhập vào chi nhánh này"));

56 │ }

57 │ else

58 │ {

59 │ return Json(new ResponseModel("Error", "Tài khoản hoặc mật khẩu không đúng"));

60 │ }

61 │ }

62 │ private string signToken(UserToken objToken)

63 │ {

64 │ string obj = Utility.Base64Encode(JsonConvert.SerializeObject(objToken));

65 │ string sign = Utility.GetSHA256(obj + QLCLConst.passcode + DateTime.Now.ToString("yyyyMM"));

66 │ string token = obj + "." + sign;

67 │ return token;

68 │ }

69 │

70 │ [HttpGet]

71 │ [TokenUserCheckHTTP(Permission.LoginAs)]

72 │ public IActionResult LoginAs(string userid)

73 │ {

74 │

75 │ DateTime now = DateTime.Now;

76 │ DateTime expiresDate;

77 │ if (now.Month == 12)

78 │ {

79 │ expiresDate = new DateTime(now.Year + 1, 1, 1);

80 │ }

81 │ else

82 │ {

83 │ expiresDate = new DateTime(now.Year, now.Month + 1, 1);

84 │ }

85 │ objToken.loginAs = true;

86 │ objToken.idAs = userid.ToInt();

87 │ string token = signToken(objToken);

88 │ Response.Cookies.Append("AuthToken", token, new() { Expires = expiresDate });

89 │ return Redirect("/home");

90 │ }

91 │ }

92 │ }

────────────────────────────────────────────────────────────

**📄 ManHinhLedController.cs (CS, 247 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.EntityFrameworkCore;

3 │ using Newtonsoft.Json;

4 │ using QuanLyChatLuong.Common;

5 │ using QuanLyChatLuong.Models;

6 │ using QuanLyChatLuong.Models.APIModels;

7 │ using QuanLyChatLuong.Models.CRMEntity;

8 │ using QuanLyChatLuong.Models.StoreProcedures;

9 │ using System.Net.Http.Headers;

10 │ using System.Text;

11 │ using System.Web.Http.ModelBinding;

12 │

13 │ namespace QuanLyChatLuong.Controllers

14 │ {

15 │ [Route("ManHinhLed")]

16 │ public class ManHinhLedController : BaseController

17 │ {

18 │ public class RequestBody

19 │ {

20 │ public string? MaChiNhanh { get; set; }

21 │ }

22 │ public class ManHinhLed

23 │ {

24 │ public string? ID { get; set; }

25 │ public string IpLed { get; set; }

26 │ public string ChiNhanh { get; set; }

27 │ public string Status { get; set; }

28 │ }

29 │ private readonly QuanLyChatLuongDbContext \_dbContext;

30 │ public ManHinhLedController(QuanLyChatLuongDbContext dbcontext,MemoryCacheManager memoryCache) : base(dbcontext,...

31 │ {

32 │ \_dbContext = dbcontext;

33 │ m\_MenuID = MenuID.manHinhLed;

34 │ }

35 │ [HttpGet]

36 │ [TokenUserCheckHTTP(Permission.View)]

37 │ public IActionResult Index()

38 │ {

39 │ var ss = dbcontext.tbl\_ManLeds.ToList();

40 │ ViewData["lstChiNhanh"] = dbcontext.tbl\_ChiNhanhs.Where(x => !(x.DaXoa ?? false) && (x.TrangThai ?? false))....

41 │ ViewData["lstManHinhLed"] = dbcontext.tbl\_ManLeds.ToList();

42 │ ViewData["CurrentBranch"] = objToken.branch;

43 │ return View();

44 │ }

45 │ [TokenUserCheckHTTP(Permission.Add)]

46 │ [HttpGet("add")]

47 │ public IActionResult Add()

48 │ {

49 │ ViewData["lstChiNhanh"] = dbcontext.tbl\_ChiNhanhs.Where(x => !(x.DaXoa ?? false) && (x.TrangThai ?? false))....

50 │ ViewData["CurrentBranch"] = objToken.branch;

51 │ return View("Form");

52 │ }

53 │ [TokenUserCheckHTTP(Permission.Edit)]

54 │ [HttpGet("edit")]

55 │ public IActionResult Edit(string id)

56 │ {

57 │ ViewData["lstChiNhanh"] = dbcontext.tbl\_ChiNhanhs.Where(x => !(x.DaXoa ?? false) && (x.TrangThai ?? false))....

58 │ ViewData["item"] = dbcontext.tbl\_ManLeds.FirstOrDefault(x => x.Ma==Cast.ToInt(id.Decrypt()));

59 │ ViewData["CurrentBranch"] = objToken.branch;

60 │ return View("Form");

61 │ }

62 │ [TokenUserCheckHTTP()]

63 │ [HttpPost("updateAPI")]

64 │ public IActionResult UpdateAPI([FromBody] ManHinhLed model)

65 │ {

66 │ var item = dbcontext.tbl\_ManLeds.FirstOrDefault(x => x.Ma == Cast.ToInt(model.ID));

67 │ if (item == null)

68 │ {

69 │ tbl\_ManLed manLed = new tbl\_ManLed();

70 │ manLed.ipLed = model.IpLed;

71 │ manLed.MaCN = Cast.ToInt(model.ChiNhanh);

72 │ manLed.TrangThai = model.Status == "1";

73 │ //dbcontext.tbl\_ManLeds.Add(item);

74 │ //dbcontext.SaveChanges();

75 │ return Json(new ResponseModel(QLCLConst.Successfully, "Cập nhật thành công"));

76 │ }

77 │

78 │ item.ipLed= model.IpLed;

79 │ item.MaCN = Cast.ToInt(model.ChiNhanh);

80 │ item.TrangThai = model.Status == "1";

81 │ try

82 │ {

83 │ //dbcontext.tbl\_ManLeds.Update(item);

84 │ //dbcontext.SaveChanges();

85 │ }

86 │ catch (Exception ex)

87 │ {

88 │ return Json(new ResponseModel(QLCLConst.Error, "Cập nhật thất bại: " + ex.Message));

89 │ }

90 │

91 │ return Json(new ResponseModel(QLCLConst.Successfully, "Cập nhật thành công"));

92 │ }

93 │ [HttpPost]

94 │ [TokenUserCheckHTTP]

95 │ public async Task<IActionResult> getData([FromBody] RequestBody obj)

96 │ {

97 │ int MaChiNhanh = Cast.ToInt(obj.MaChiNhanh) == 0 ? objToken.branch : Cast.ToInt(obj.MaChiNhanh);

98 │ var content = dbcontext.Set<sp\_HienThiLedResult>().FromSqlRaw($"sp\_HienThiLed1 @MaChiNhanh={MaChiNhanh}").To...

99 │ return Json(new ResponseModel(QLCLConst.OK, "", content));

100 │

101 │ }

102 │ Dictionary<string, int> HexToRgb(string hex)

103 │ {

104 │ return new Dictionary<string, int>

105 │ {

106 │ ["R"] = Convert.ToInt32(hex.Substring(1, 2), 16),

107 │ ["G"] = Convert.ToInt32(hex.Substring(3, 2), 16),

108 │ ["B"] = Convert.ToInt32(hex.Substring(5, 2), 16),

109 │ };

110 │ }

111 │

112 │ [HttpPost("getDataManHinhLed")]

113 │ [TokenUserCheckHTTP(Permission.NotCheck)]

114 │ public async Task<IActionResult> getDataManHinhLed([FromBody] RequestBody obj)

115 │ {

116 │ int MaChiNhanhX = Cast.ToInt(obj.MaChiNhanh) == 0?objToken.branch: Cast.ToInt(obj.MaChiNhanh);

117 │ var content = dbcontext.Set<sp\_HienThiLedResult>().FromSqlRaw($"sp\_HienThiLed1 @MaChiNhanh={MaChiNhanhX}").T...

118 │

119 │ //string hostAPILed\_NM5 = "http://172.16.10.54:8000/docs";

120 │ tbl\_ManLed ManHinhLedChiNhanh = new tbl\_ManLedBusiness(dbcontext).GetDataManLed(MaChiNhanhX).FirstOrDefault(...

121 │ string sss = JsonConvert.SerializeObject(content);

122 │ if (ManHinhLedChiNhanh != null)

123 │ {

124 │ HttpRequestMessage httpRequestMessage = null;

125 │ string responseString = string.Empty;

126 │ if (string.IsNullOrWhiteSpace(ManHinhLedChiNhanh.ipLed))

127 │ {

128 │ throw new ArgumentException("URL không được để trống.");

129 │ }

130 │

131 │ string url = ManHinhLedChiNhanh.ipLed;

132 │

133 │ if (!url.StartsWith("http://") && !url.StartsWith("https://"))

134 │ {

135 │ url = "http://" + url;

136 │ }

137 │ string stringContent = "";

138 │ if (MaChiNhanhX == (int) MaChiNhanh.NM5)

139 │ {

140 │ List<tbl\_ColorGioiHanSoLieu> color = dbcontext.tbl\_ColorGioiHanSoLieus.Where(x => x.MaChiNhanh == Ma...

141 │ List<tbl\_GioiHanSoLieu> gioihan = dbcontext.tbl\_GioiHanSoLieus.Where(x => x.MaChiNhanh == (int)MaChi...

142 │ var solieu2 = content.FirstOrDefault();

143 │ var redColor = HexToRgb("#FF0000");

144 │ var blueColor = HexToRgb("#0000FF");

145 │ var whiteColor = HexToRgb("#FFFFFF");

146 │ var DoTrangColor = whiteColor;

147 │ var DoTrangGHMaxColor = gioihan.FirstOrDefault(x => x.Code == "B\_DoTrang" && x.GiaTri <= Cast.ToDeci...

148 │ if (DoTrangGHMaxColor != null)

149 │ {

150 │ var color1 = color.FirstOrDefault(x => x.Code == "B\_DoTrang");

151 │ if (color1 != null && color1.MaxColor != null) DoTrangColor = HexToRgb(color1.MaxColor);

152 │ }

153 │ var DoTrangGHMinColor = gioihan.FirstOrDefault(x => x.Code == "B\_DoTrang" && x.GiaTri >= Cast.ToDeci...

154 │ if (DoTrangGHMinColor != null)

155 │ {

156 │ var color1 = color.FirstOrDefault(x => x.Code == "B\_DoTrang");

157 │ if (color1 != null && color1.MinColor != null) DoTrangColor = HexToRgb(color1.MinColor);

158 │ }

159 │ var DoAmColor = whiteColor;

160 │ var DoAmGHMaxColor = gioihan.FirstOrDefault(x => x.Code == "B\_DoAm" && x.GiaTri <= Cast.ToDecimal(so...

161 │ if (DoAmGHMaxColor != null)

162 │ {

163 │ var color1 = color.FirstOrDefault(x => x.Code == "B\_DoAm");

164 │ if (color1 != null && color1.MaxColor != null) DoAmColor = HexToRgb(color1.MaxColor);

165 │ }

166 │ var DoAmGHMinColor = gioihan.FirstOrDefault(x => x.Code == "B\_DoAm" && x.GiaTri >= Cast.ToDecimal(so...

167 │ if (DoAmGHMinColor != null)

168 │ {

169 │ var color1 = color.FirstOrDefault(x => x.Code == "B\_DoAm");

170 │ if (color1 != null && color1.MinColor != null) DoAmColor = HexToRgb(color1.MinColor);

171 │ }

172 │ var TapChatColor = whiteColor;

173 │ var TapChatGHMaxColor = gioihan.FirstOrDefault(x => x.Code == "B\_TapChat" && x.GiaTri <= Cast.ToDeci...

174 │ if (TapChatGHMaxColor != null)

175 │ {

176 │ var color1 = color.FirstOrDefault(x => x.Code == "B\_TapChat");

177 │ if (color1 != null && color1.MaxColor != null) TapChatColor = HexToRgb(color1.MaxColor);

178 │ }

179 │ var TapChatGHMinColor = gioihan.FirstOrDefault(x => x.Code == "B\_TapChat" && x.GiaTri >= Cast.ToDeci...

180 │ if (TapChatGHMinColor != null)

181 │ {

182 │ var color1 = color.FirstOrDefault(x => x.Code == "B\_TapChat");

183 │ if (color1 != null && color1.MinColor != null) TapChatColor = HexToRgb(color1.MinColor);

184 │ }

185 │ var objSend = new

186 │ {

187 │

188 │ do\_trang = solieu2.B\_DoTrang,

189 │ do\_trang\_color = DoTrangColor,

190 │ do\_am = solieu2.B\_DoAm,

191 │ do\_am\_color = DoAmColor,

192 │ tap\_chat = solieu2.B\_TapChat,

193 │ tap\_chat\_color = TapChatColor

194 │ };

195 │ stringContent = JsonConvert.SerializeObject(objSend);

196 │ }

197 │ else

198 │ {

199 │ stringContent = JsonConvert.SerializeObject(content);

200 │ }

201 │ if (MaChiNhanhX == (int)MaChiNhanh.NM2)

202 │ {

203 │ httpRequestMessage = new HttpRequestMessage

204 │ {

205 │ Method = HttpMethod.Get,

206 │ };

207 │ httpRequestMessage.Headers.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

208 │ var solieu = content.FirstOrDefault();

209 │ url += "?dt=" + solieu.B\_DoTrang + "&da=" + solieu.B\_DoAm + "&tc=" + solieu.B\_TapChat;

210 │ }

211 │ else

212 │ {

213 │ httpRequestMessage = new HttpRequestMessage

214 │ {

215 │ Method = HttpMethod.Post,

216 │ Content = new StringContent(stringContent, Encoding.UTF8, "application/json"),

217 │ };

218 │

219 │ httpRequestMessage.Headers.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

220 │ }

221 │

222 │ try

223 │ {

224 │ Uri uri = new Uri(url);

225 │ using (HttpClient client = new HttpClient() { BaseAddress = uri })

226 │ {

227 │ HttpResponseMessage response = await client.SendAsync(httpRequestMessage);

228 │ responseString = await response.Content.ReadAsStringAsync();

229 │ }

230 │ if (responseString.Contains("Not found"))

231 │ {

232 │ return Json(new ResponseModel(QLCLConst.Error, "Không tìm thấy màn hình led", responseStri...

233 │ }

234 │ return Json(new ResponseModel(QLCLConst.OK, "Gửi dữ liệu thành công", responseString));

235 │ }

236 │ catch (Exception e)

237 │ {

238 │ return Json(new ResponseModel(QLCLConst.Error, "Gửi dữ liệu không thành công"));

239 │ }

240 │ }

241 │ else

242 │ {

243 │ return Json(new ResponseModel(QLCLConst.Error, "Xem lai man hinh led"));

244 │ }

245 │ }

246 │ }

247 │ }

────────────────────────────────────────────────────────────

**📄 MemoryCacheManager.cs (CS, 33 lines)**

1 │ ﻿using Microsoft.Extensions.Caching.Memory;

2 │ using Microsoft.Extensions.Options;

3 │

4 │ namespace QuanLyChatLuong.Common

5 │ {

6 │ public sealed class MemoryCacheManager : IMemoryCache

7 │ {

8 │ readonly IOptions<MemoryCacheOptions> \_optionsAccessor;

9 │

10 │ IMemoryCache \_inner;

11 │

12 │ public MemoryCacheManager(IOptions<MemoryCacheOptions> optionsAccessor)

13 │ {

14 │ \_optionsAccessor = optionsAccessor;

15 │ \_inner = new MemoryCache(\_optionsAccessor);

16 │ }

17 │

18 │ public void Reset()

19 │ {

20 │ var old = \_inner;

21 │ \_inner = new MemoryCache(\_optionsAccessor);

22 │ old.Dispose();

23 │ }

24 │

25 │ public void Dispose() => \_inner.Dispose();

26 │

27 │ public ICacheEntry CreateEntry(object key) => \_inner.CreateEntry(key);

28 │

29 │ public void Remove(object key) => \_inner.Remove(key);

30 │

31 │ public bool TryGetValue(object key, out object value) => \_inner.TryGetValue(key, out value);

32 │ }

33 │ }

────────────────────────────────────────────────────────────

**📄 MenuController.cs (CS, 267 lines)**

1 │ ﻿using Microsoft.AspNetCore.Authentication;

2 │ using Microsoft.AspNetCore.Authentication.Cookies;

3 │ using Microsoft.AspNetCore.Mvc;

4 │ using Microsoft.Extensions.Logging;

5 │ using Microsoft.IdentityModel.Tokens;

6 │ using System.IdentityModel.Tokens.Jwt;

7 │ using System.Security.Claims;

8 │ using System.Text;

9 │ using System.Threading.Tasks;

10 │ using Microsoft.EntityFrameworkCore;

11 │ using QuanLyChatLuong.Models;

12 │ using System.Security.Cryptography;

13 │ using QuanLyChatLuong.Common;

14 │ using QuanLyChatLuong.Models.APIModels;

15 │

16 │ namespace QuanLyChatLuong.Controllers

17 │ {

18 │ [Route("sysMenu")]

19 │ public class MenuController : BaseController

20 │ {

21 │ public MenuController(QuanLyChatLuongDbContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext, memo...

22 │ {

23 │ m\_MenuID = MenuID.sysMenu;

24 │ }

25 │ [TokenUserCheckHTTP(Permission.View)]

26 │ [HttpGet]

27 │ public IActionResult Index()

28 │ {

29 │

30 │ return View();

31 │ }

32 │ private bool CheckExists(int Ma, string TenSanPham)

33 │ {

34 │ return dbcontext.sys\_Menus.Any(x => x.Ten == TenSanPham && x.Ma != Ma && !(x.DaXoa ?? false));

35 │ }

36 │

37 │

38 │ private ResponseModel CheckRequire(string Ten, string Code, string duongDan, string soThuTu)

39 │ {

40 │ if (string.IsNullOrEmpty(Ten))

41 │ {

42 │ return new ResponseModel(QLCLConst.Error, "Chưa nhập tên");

43 │ }

44 │ if (string.IsNullOrEmpty(Code))

45 │ {

46 │ return new ResponseModel(QLCLConst.Error, "Chưa nhập mã code");

47 │ }

48 │ if (string.IsNullOrEmpty(soThuTu))

49 │ {

50 │ return new ResponseModel(QLCLConst.Error, "Chưa nhập số thứ tự");

51 │ }

52 │ return null;

53 │ }

54 │

55 │ [TokenUserCheckHTTP]

56 │ [HttpGet("getListMenu")]

57 │ public IActionResult getListMenu()

58 │ {

59 │ var listData = dbcontext.sys\_Menus

60 │ .Where(x => !(x.DaXoa ?? false))

61 │ .Select(x => new

62 │ {

63 │ x.Ma,

64 │ eMa = x.Ma.Encrypt(),

65 │ x.Code,

66 │ x.MenuCha,

67 │ x.Ten,

68 │ x.DuongDan,

69 │ x.Icon,

70 │ x.IconWeb,

71 │ x.SoThuTu,

72 │ x.ThamSo,

73 │ x.HienMenu,

74 │ x.MenuCon,

75 │ LuonLuonHien = x.LuonLuonHien ?? false,

76 │ x.TrangThai,

77 │ x.NguoiTao,

78 │ x.NgayTao,

79 │ x.NguoiSua,

80 │ x.NgaySua,

81 │ x.DaXoa,

82 │ x.NguoiXoa,

83 │ x.NgayXoa,

84 │ MenuChaTen = dbcontext.sys\_Menus

85 │ .Where(m => m.Ma == x.MenuCha)

86 │ .Select(m => m.Ten)

87 │ .FirstOrDefault(),

88 │ MenuConTen = dbcontext.sys\_Menus

89 │ .Where(m => m.Ma == x.MenuCon)

90 │ .Select(m => m.Ten)

91 │ .FirstOrDefault()

92 │ })

93 │ .ToList();

94 │

95 │ return Json(new { success = true, listData });

96 │ }

97 │

98 │ [TokenUserCheckHTTP(Permission.Add)]

99 │ [HttpGet("add")]

100 │ public IActionResult Add()

101 │ {

102 │ var listMenuCha = dbcontext.sys\_Menus

103 │ .Where(x => !(x.DaXoa ?? false) && x.MenuCha == null)

104 │ .Select(x => new sys\_Menu

105 │ {

106 │ Ma = x.Ma,

107 │ Ten = x.Ten

108 │ })

109 │ .ToList();

110 │ var listMenuCon = dbcontext.sys\_Menus

111 │ .Where(x => !(x.DaXoa ?? false) && x.MenuCha != null)

112 │ .Select(x => new sys\_Menu

113 │ {

114 │ Ma = x.Ma,

115 │ Ten = x.Ten

116 │ })

117 │ .ToList();

118 │

119 │ ViewData["listMenuCha"] = listMenuCha;

120 │ ViewData["listMenuCon"] = listMenuCon;

121 │

122 │ return View("Form");

123 │ }

124 │

125 │ [TokenUserCheckHTTP()]

126 │ [HttpPost("addAPI")]

127 │ public IActionResult AddAPI(int Ma, string Ten, string Code, string duongDan, string Icon, string soThuTu, strin...

128 │ {

129 │ var result = CheckRequire(Ten, Code, duongDan, soThuTu);

130 │ if (result != null)

131 │ {

132 │ return Json(result);

133 │ }

134 │

135 │ if (CheckExists(Ma, Ten))

136 │ {

137 │ return Json(new ResponseModel(QLCLConst.Error, "Tên menu đã tồn tại"));

138 │ }

139 │

140 │ var item = new sys\_Menu

141 │ {

142 │ Ten = Ten,

143 │ Code = Code,

144 │ DuongDan = duongDan,

145 │ IconWeb = Icon,

146 │ SoThuTu = Cast.ToInt(soThuTu),

147 │ MenuCha = Cast.ToInt(menuCha),

148 │ MenuCon = Cast.ToInt(menuCon),

149 │ TrangThai = TrangThai == "1",

150 │ NguoiTao = objToken!.userid,

151 │ NgayTao = DateTime.Now

152 │ };

153 │

154 │

155 │ dbcontext.sys\_Menus.Add(item);

156 │ dbcontext.SaveChanges();

157 │

158 │ return Json(new ResponseModel(QLCLConst.Successfully, "Thêm thành công"));

159 │ }

160 │

161 │ [TokenUserCheckHTTP(Permission.Edit)]

162 │ [HttpGet("edit")]

163 │ public IActionResult Edit(string id)

164 │ {

165 │ var item = dbcontext.sys\_Menus.FirstOrDefault(c => c.Ma == Cast.ToInt(id.Decrypt()));

166 │

167 │ if (item == null)

168 │ {

169 │ return NotFound("Menu not found");

170 │ }

171 │

172 │ var listMenuCha = dbcontext.sys\_Menus

173 │ .Where(x => !(x.DaXoa ?? false) && (x.MenuCha == 0 || x.MenuCha == null) && (x.HienMenu ??false))

174 │ .Select(x => new sys\_Menu

175 │ {

176 │ Ma = x.Ma,

177 │ Ten = x.Ten

178 │ })

179 │ .ToList();

180 │

181 │ var listMenuCon = dbcontext.sys\_Menus

182 │ .Where(x => !(x.DaXoa ?? false) && x.MenuCha != null)

183 │ .Select(x => new sys\_Menu

184 │ {

185 │ Ma = x.Ma,

186 │ Ten = x.Ten

187 │ })

188 │ .ToList();

189 │

190 │ ViewData["item"] = item;

191 │ ViewData["listMenuCha"] = listMenuCha;

192 │ ViewData["listMenuCon"] = listMenuCon;

193 │

194 │ return View("Form");

195 │ }

196 │ [TokenUserCheckHTTP()]

197 │ [HttpPost("updateAPI")]

198 │ public IActionResult UpdateAPI(int Ma, string Ten, string Code, string duongDan, string Icon, string soThuTu, st...

199 │ {

200 │ var item = dbcontext.sys\_Menus.FirstOrDefault(x => x.Ma == Ma && !(x.DaXoa ?? false));

201 │ if (item == null)

202 │ {

203 │ return Json(new ResponseModel(QLCLConst.Error, "Menu không tồn tại"));

204 │ }

205 │ if (dbcontext.sys\_Menus.Any(x => x.Ten == Ten && x.Ma != Ma))

206 │ {

207 │ return Json(new ResponseModel(QLCLConst.Error, "Tên sản phẩm đã tồn tại"));

208 │ }

209 │

210 │ item.Ten = Ten;

211 │ item.Code = Code;

212 │ item.DuongDan = duongDan;

213 │ item.IconWeb = Icon;

214 │ item.SoThuTu = Cast.ToInt(soThuTu);

215 │ item.MenuCha = Cast.ToInt(menuCha);

216 │ item.MenuCon = Cast.ToInt(menuCon);

217 │ item.TrangThai = TrangThai == "1";

218 │ item.NguoiSua = objToken!.userid;

219 │ item.NgaySua = DateTime.Now;

220 │ try

221 │ {

222 │ dbcontext.sys\_Menus.Update(item);

223 │ dbcontext.SaveChanges();

224 │ }

225 │ catch (Exception ex)

226 │ {

227 │ return Json(new ResponseModel(QLCLConst.Error, "Cập nhật thất bại: " + ex.Message));

228 │ }

229 │

230 │ return Json(new ResponseModel(QLCLConst.Successfully, "Cập nhật thành công"));

231 │ }

232 │

233 │ [HttpPost("DeleteAPI")]

234 │ [TokenUserCheckHTTP(Permission.Delete)]

235 │ public IActionResult DeleteAPI(string eMa)

236 │ {

237 │ var item = dbcontext.sys\_Menus.FirstOrDefault(x => x.Ma == Cast.ToInt(eMa.Decrypt()));

238 │ if (item == null)

239 │ {

240 │ return Json(QLCLConst.ResponseNotExist);

241 │ }

242 │ item.DaXoa = true;

243 │ dbcontext.SaveChanges();

244 │ return Json(QLCLConst.ResponseDeleteSuccessfully);

245 │ }

246 │ [TokenUserCheckHTTP()]

247 │ [HttpPost("getListCheckBox")]

248 │ public IActionResult getListCheckBox(string isChecked, string ma)

249 │ {

250 │ var item = dbcontext.sys\_Menus.FirstOrDefault(x => x.Ma == Cast.ToInt(ma));

251 │

252 │ if (item == null)

253 │ {

254 │ return Json(new ResponseModel("Error", "Menu không tồn tại"));

255 │ }

256 │

257 │ item.HienMenu = Cast.ToBoolean(isChecked);

258 │ item.NguoiSua = objToken!.userid;

259 │ item.NgaySua = DateTime.Now;

260 │

261 │ dbcontext.SaveChanges();

262 │ return Json(new ResponseModel(QLCLConst.Successfully, "Chỉnh sửa thành công"));

263 │ }

264 │

265 │

266 │ }

267 │ }

────────────────────────────────────────────────────────────

**📄 MesThongBaoApp.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class MesThongBaoApp

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaUser { get; set; }

11 │

12 │ public string? TieuDe { get; set; }

13 │

14 │ public string? NoiDung { get; set; }

15 │

16 │ public bool? DaXem { get; set; }

17 │

18 │ public int? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public int? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public int? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 MultiSQL.cs (CS, 12 lines)**

1 │ ﻿

2 │

3 │ namespace QuanLyChatLuong.Models

4 │ {

5 │ public partial class QuanLyChatLuongDbContext

6 │ {

7 │ public int? userId { get; set; }

8 │

9 │ }

10 │ }

11 │ //Multi SQL ID Generator

12 │ //ConcurrentStack<long> ids = new ConcurrentStack<long>();

────────────────────────────────────────────────────────────

**📄 MultiServer.cs (CS, 24 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using System.Net.Sockets;

3 │ using System.Net;

4 │ using System.Text;

5 │ using Newtonsoft.Json;

6 │ using Microsoft.EntityFrameworkCore.Metadata.Internal;

7 │ using Microsoft.AspNetCore.Mvc;

8 │

9 │ using System.Data;

10 │ using System.Reflection;

11 │ using Microsoft.EntityFrameworkCore.Infrastructure;

12 │ using System.ComponentModel;

13 │ using System.Diagnostics.CodeAnalysis;

14 │ using System;

15 │ using System.Diagnostics;

16 │ using static Microsoft.EntityFrameworkCore.DbLoggerCategory;

17 │

18 │ namespace QuanLyChatLuong.Common.MultiServer

19 │ {

20 │ public class MultiServer : Controller

21 │ {

22 │

23 │ }

24 │ }

────────────────────────────────────────────────────────────

**📄 Nhập.cs (CS, 13 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class Nhập

7 │ {

8 │ public string? SốPhiếu { get; set; }

9 │

10 │ public string? NhàCungCấpMới { get; set; }

11 │

12 │ public string? Mancc { get; set; }

13 │ }

────────────────────────────────────────────────────────────

**📄 NotEqual.cs (CS, 24 lines)**

1 │ ﻿namespace QuanLyChatLuong.Common

2 │ {

3 │ public class NotEqual : IRouteConstraint

4 │ {

5 │ private string \_match = String.Empty;

6 │

7 │ public NotEqual(string match)

8 │ {

9 │ \_match = match;

10 │ }

11 │

12 │ public bool Match(HttpContext? httpContext, IRouter? route, string parameterName, RouteValueDictionary values, R...

13 │ {

14 │ try

15 │ {

16 │ return String.Compare(values[parameterName]!.ToString(), \_match, true) != 0;

17 │ }

18 │ catch

19 │ {

20 │ return false;

21 │ }

22 │ }

23 │ }

24 │ }

────────────────────────────────────────────────────────────

**📄 NtfGroupThongBao.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class NtfGroupThongBao

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? Status { get; set; }

15 │

16 │ public int? UserCreated { get; set; }

17 │

18 │ public DateTime? CreatedDate { get; set; }

19 │

20 │ public int? UserModified { get; set; }

21 │

22 │ public DateTime? ModifiedDate { get; set; }

23 │

24 │ public bool? Deleted { get; set; }

25 │

26 │ public int? UserDeleted { get; set; }

27 │

28 │ public DateTime? DeletedDate { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 NtfMemberGroupThongBao.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class NtfMemberGroupThongBao

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaGroupThongBao { get; set; }

11 │

12 │ public int? MaUser { get; set; }

13 │

14 │ public int? UserCreated { get; set; }

15 │

16 │ public DateTime? CreatedDate { get; set; }

17 │

18 │ public int? UserModified { get; set; }

19 │

20 │ public DateTime? ModifiedDate { get; set; }

21 │

22 │ public bool? Deleted { get; set; }

23 │

24 │ public int? UserDeleted { get; set; }

25 │

26 │ public DateTime? DeletedDate { get; set; }

27 │ }

────────────────────────────────────────────────────────────

**📄 OpsBaoGia.cs (CS, 75 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class OpsBaoGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaChiNhanh { get; set; }

11 │

12 │ public string? SoBaoGia { get; set; }

13 │

14 │ public int? MaKh { get; set; }

15 │

16 │ public int? MaNhaVanChuyen { get; set; }

17 │

18 │ public int? NgayCongNo { get; set; }

19 │

20 │ public DateTime? NgayBaoGia { get; set; }

21 │

22 │ public DateTime? TuNgay { get; set; }

23 │

24 │ public DateTime? DenNgay { get; set; }

25 │

26 │ public string? ThoiGianGiaoHang { get; set; }

27 │

28 │ public string? DiaDiemGiaoHang { get; set; }

29 │

30 │ public string? ThoiGianThanhToan { get; set; }

31 │

32 │ public string? GhiChu { get; set; }

33 │

34 │ public int? NguoiLienHe { get; set; }

35 │

36 │ public bool? ChotHopDong { get; set; }

37 │

38 │ public int? MaTrangThai { get; set; }

39 │

40 │ public int? MaTienTe { get; set; }

41 │

42 │ public int? MaTyGia { get; set; }

43 │

44 │ public int? TyGia { get; set; }

45 │

46 │ public int? MaNvphuTrach { get; set; }

47 │

48 │ public int? MaNvhoTro { get; set; }

49 │

50 │ public bool? RemindExpireDate { get; set; }

51 │

52 │ public bool? RemindSendReport { get; set; }

53 │

54 │ public int? TieuChuanChatLuong { get; set; }

55 │

56 │ public string? ChiPhiBocXep { get; set; }

57 │

58 │ public int? SoTanMoiContainer { get; set; }

59 │

60 │ public int? LoaiContainer { get; set; }

61 │

62 │ public int? UserCreated { get; set; }

63 │

64 │ public DateTime? CreatedDate { get; set; }

65 │

66 │ public int? UserModified { get; set; }

67 │

68 │ public DateTime? ModifiedDate { get; set; }

69 │

70 │ public bool? Deleted { get; set; }

71 │

72 │ public int? UserDeleted { get; set; }

73 │

74 │ public DateTime? DeletedDate { get; set; }

75 │ }

────────────────────────────────────────────────────────────

**📄 OpsChiTietBaoGia.cs (CS, 67 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class OpsChiTietBaoGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaBaoGia { get; set; }

11 │

12 │ public int? MaHh { get; set; }

13 │

14 │ public int? MaDvt { get; set; }

15 │

16 │ public int? LoaiBao { get; set; }

17 │

18 │ public int? SoLuong { get; set; }

19 │

20 │ public int? SoLuongDaXuat { get; set; }

21 │

22 │ public int? MaGiaNiemYet { get; set; }

23 │

24 │ public decimal? DonGiaTaiKho { get; set; }

25 │

26 │ public decimal? BaoBiHd { get; set; }

27 │

28 │ public decimal? Bao10Kg { get; set; }

29 │

30 │ public decimal? GiaVanChuyen { get; set; }

31 │

32 │ public decimal? BocXep { get; set; }

33 │

34 │ public int? NgayCongNo { get; set; }

35 │

36 │ public decimal? ChiPhiNgayCongNo { get; set; }

37 │

38 │ public decimal? HoaHong { get; set; }

39 │

40 │ public decimal? ChiPhiHoTro { get; set; }

41 │

42 │ public decimal? DonGiaSauTinh { get; set; }

43 │

44 │ public decimal? DonGiaBan { get; set; }

45 │

46 │ public decimal? ThanhTien { get; set; }

47 │

48 │ public int? MaLyDo { get; set; }

49 │

50 │ public string? GhiChu { get; set; }

51 │

52 │ public bool? Status { get; set; }

53 │

54 │ public int? UserCreated { get; set; }

55 │

56 │ public DateTime? CreatedDate { get; set; }

57 │

58 │ public int? UserModified { get; set; }

59 │

60 │ public DateTime? ModifiedDate { get; set; }

61 │

62 │ public bool? Deleted { get; set; }

63 │

64 │ public int? UserDeleted { get; set; }

65 │

66 │ public DateTime? DeletedDate { get; set; }

67 │ }

────────────────────────────────────────────────────────────

**📄 OpsChiTietDoanhThu.cs (CS, 75 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class OpsChiTietDoanhThu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaDoanhThu { get; set; }

11 │

12 │ public int? MaHang { get; set; }

13 │

14 │ public int? MaDvt { get; set; }

15 │

16 │ public string? KhoHang { get; set; }

17 │

18 │ public string? Lo { get; set; }

19 │

20 │ public int? SoLuong { get; set; }

21 │

22 │ public int? SoLuongThucXuat { get; set; }

23 │

24 │ public decimal? GiaBan { get; set; }

25 │

26 │ public decimal? ThanhTien { get; set; }

27 │

28 │ public decimal? ChietKhau { get; set; }

29 │

30 │ public int? MaThue { get; set; }

31 │

32 │ public decimal? ThueVat { get; set; }

33 │

34 │ public decimal? TienThue { get; set; }

35 │

36 │ public decimal? PhaiThu { get; set; }

37 │

38 │ public int? LoaiBao { get; set; }

39 │

40 │ public decimal? DonGiaTaiKho { get; set; }

41 │

42 │ public decimal? BaoBiHd { get; set; }

43 │

44 │ public decimal? Bao10Kg { get; set; }

45 │

46 │ public decimal? GiaVanChuyen { get; set; }

47 │

48 │ public decimal? GiaVonVanChuyen { get; set; }

49 │

50 │ public decimal? BocXep { get; set; }

51 │

52 │ public int? NgayCongNo { get; set; }

53 │

54 │ public decimal? ChiPhiNgayCongNo { get; set; }

55 │

56 │ public decimal? HoaHong { get; set; }

57 │

58 │ public decimal? ChiPhiHoTro { get; set; }

59 │

60 │ public decimal? DonGiaSauTinh { get; set; }

61 │

62 │ public decimal? GiaNiemYet { get; set; }

63 │

64 │ public int? MaLyDo { get; set; }

65 │

66 │ public decimal? GiaTaiKhoCoBaoBi { get; set; }

67 │

68 │ public int? MaHopDongCt { get; set; }

69 │

70 │ public int? SoLuongHd { get; set; }

71 │

72 │ public int? SoLuongConLai { get; set; }

73 │

74 │ public decimal? GiaGiam { get; set; }

75 │ }

────────────────────────────────────────────────────────────

**📄 OpsDoanhThu.cs (CS, 93 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class OpsDoanhThu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaChiNhanh { get; set; }

11 │

12 │ public int? MaPhongBan { get; set; }

13 │

14 │ public DateTime? NgayChungTu { get; set; }

15 │

16 │ public int? MaHopDong { get; set; }

17 │

18 │ public string? SoHopDong { get; set; }

19 │

20 │ public string? SoChungTu { get; set; }

21 │

22 │ public int? MaKh { get; set; }

23 │

24 │ public int? MaNvbh { get; set; }

25 │

26 │ public int? MaNhaVanChuyen { get; set; }

27 │

28 │ public string? NhaVanChuyen { get; set; }

29 │

30 │ public int? MaChiNhanhXuat { get; set; }

31 │

32 │ public int? MaKho { get; set; }

33 │

34 │ public string? DienGiai { get; set; }

35 │

36 │ public DateTime? NgayHetHanThanhToan { get; set; }

37 │

38 │ public int? MaTienTe { get; set; }

39 │

40 │ public int? MaTyGia { get; set; }

41 │

42 │ public int? TyGia { get; set; }

43 │

44 │ public string? SoXe { get; set; }

45 │

46 │ public string? SoPo { get; set; }

47 │

48 │ public bool? DaXuat { get; set; }

49 │

50 │ public int? TrangThaiDonHang { get; set; }

51 │

52 │ public bool? XuatKhau { get; set; }

53 │

54 │ public bool? DonHangChoThanhToan { get; set; }

55 │

56 │ public decimal? DaThanhToan { get; set; }

57 │

58 │ public decimal? ConLai { get; set; }

59 │

60 │ public int? UserCreated { get; set; }

61 │

62 │ public DateTime? CreatedDate { get; set; }

63 │

64 │ public int? UserModified { get; set; }

65 │

66 │ public DateTime? ModifiedDate { get; set; }

67 │

68 │ public bool? Deleted { get; set; }

69 │

70 │ public int? UserDeleted { get; set; }

71 │

72 │ public DateTime? DeletedDate { get; set; }

73 │

74 │ public bool? Exported { get; set; }

75 │

76 │ public int? UserExportd { get; set; }

77 │

78 │ public DateTime? ExportedDate { get; set; }

79 │

80 │ public int? MaDonHangTraLai { get; set; }

81 │

82 │ public DateTime? NgayKetThuc { get; set; }

83 │

84 │ public int? MaCnxuatHoaDon { get; set; }

85 │

86 │ public int? MaLyDoDuyet { get; set; }

87 │

88 │ public string? LyDoDuyet { get; set; }

89 │

90 │ public int? MaNguoiLienHe { get; set; }

91 │

92 │ public DateTime? NgayDuKienGiaoHang { get; set; }

93 │ }

────────────────────────────────────────────────────────────

**📄 OpsLichSuChiTietBaoGia.cs (CS, 67 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class OpsLichSuChiTietBaoGia

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiTiet { get; set; }

11 │

12 │ public int? MaBaoGia { get; set; }

13 │

14 │ public int? MaHh { get; set; }

15 │

16 │ public int? LoaiBao { get; set; }

17 │

18 │ public int? SoLuong { get; set; }

19 │

20 │ public int? SoLuongDaXuat { get; set; }

21 │

22 │ public int? MaGiaNiemYet { get; set; }

23 │

24 │ public int? DonGiaTaiKho { get; set; }

25 │

26 │ public int? BaoBiHd { get; set; }

27 │

28 │ public int? Bao10Kg { get; set; }

29 │

30 │ public int? GiaVanChuyen { get; set; }

31 │

32 │ public int? BocXep { get; set; }

33 │

34 │ public int? NgayCongNo { get; set; }

35 │

36 │ public int? ChiPhiNgayCongNo { get; set; }

37 │

38 │ public int? HoaHong { get; set; }

39 │

40 │ public int? ChiPhiHoTro { get; set; }

41 │

42 │ public int? DonGiaSauTinh { get; set; }

43 │

44 │ public int? DonGiaBan { get; set; }

45 │

46 │ public decimal? ThanhTien { get; set; }

47 │

48 │ public int? MaLyDo { get; set; }

49 │

50 │ public string? GhiChu { get; set; }

51 │

52 │ public bool? Status { get; set; }

53 │

54 │ public int? UserCreated { get; set; }

55 │

56 │ public DateTime? CreatedDate { get; set; }

57 │

58 │ public int? UserModified { get; set; }

59 │

60 │ public DateTime? ModifiedDate { get; set; }

61 │

62 │ public bool? Deleted { get; set; }

63 │

64 │ public int? UserDeleted { get; set; }

65 │

66 │ public DateTime? DeletedDate { get; set; }

67 │ }

────────────────────────────────────────────────────────────

**📄 OpsThanhToan.cs (CS, 41 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class OpsThanhToan

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaChiNhanh { get; set; }

11 │

12 │ public int? MaDoanhThu { get; set; }

13 │

14 │ public string? SoChungTu { get; set; }

15 │

16 │ public int? MaKh { get; set; }

17 │

18 │ public string? MaKh1 { get; set; }

19 │

20 │ public DateTime? NgayThanhToan { get; set; }

21 │

22 │ public decimal? SoTien { get; set; }

23 │

24 │ public string? GhiChu { get; set; }

25 │

26 │ public bool? Status { get; set; }

27 │

28 │ public int? UserCreated { get; set; }

29 │

30 │ public DateTime? CreatedDate { get; set; }

31 │

32 │ public int? UserModified { get; set; }

33 │

34 │ public DateTime? ModifiedDate { get; set; }

35 │

36 │ public bool? Deleted { get; set; }

37 │

38 │ public int? UserDeleted { get; set; }

39 │

40 │ public DateTime? DeletedDate { get; set; }

41 │ }

────────────────────────────────────────────────────────────

**📄 OpsTieuChuanKyThuat.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models.CRMEntity;

5 │

6 │ public partial class OpsTieuChuanKyThuat

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaBaoGia { get; set; }

11 │

12 │ public int? MaTieuChuan { get; set; }

13 │

14 │ public string? TieuChuan { get; set; }

15 │

16 │ public bool? Status { get; set; }

17 │

18 │ public int? UserCreated { get; set; }

19 │

20 │ public DateTime? CreatedDate { get; set; }

21 │

22 │ public int? UserModified { get; set; }

23 │

24 │ public DateTime? ModifiedDate { get; set; }

25 │

26 │ public bool? Deleted { get; set; }

27 │

28 │ public int? UserDeleted { get; set; }

29 │

30 │ public DateTime? DeletedDate { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 PageNotFoundController.cs (CS, 27 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using QuanLyChatLuong.Common;

3 │

4 │ namespace QuanLyChatLuong.Controllers

5 │ {

6 │ public class PageNotFoundController : Controller

7 │ {

8 │ public ActionResult Index()

9 │ {

10 │ return View();

11 │ }

12 │

13 │ public ActionResult BaoTri()

14 │ {

15 │ return View();

16 │ }

17 │ public ActionResult LoginAsDenied()

18 │ {

19 │ return View();

20 │ }

21 │

22 │ public ActionResult PermissionDenied()

23 │ {

24 │ return View();

25 │ }

26 │ }

27 │ }

────────────────────────────────────────────────────────────

**📄 PermissionController.cs (CS, 105 lines)**

1 │ ﻿using Microsoft.AspNetCore.Mvc;

2 │ using Microsoft.AspNetCore.Mvc.ModelBinding;

3 │ using QuanLyChatLuong.Common;

4 │ using QuanLyChatLuong.Models;

5 │ using QuanLyChatLuong.Models.APIModels;

6 │ using System.Net.WebSockets;

7 │ using System.Runtime.CompilerServices;

8 │

9 │ namespace QuanLyChatLuong.Controllers

10 │ {

11 │ [Route("sysQuyen")]

12 │ public class PermissionController : BaseController

13 │ {

14 │ public PermissionController(QuanLyChatLuongDbContext dbcontext, MemoryCacheManager memoryCache) : base(dbcontext...

15 │ {

16 │ m\_MenuID = MenuID.sysQuyen;

17 │ }

18 │ public class PermissionForm

19 │ {

20 │ public string? ID { get; set; }

21 │ public string? Code { get; set; }

22 │ public string? Name { get; set; }

23 │ public string? Description { get; set; }

24 │ public string? FormAccess { get; set; }

25 │ public string? Status { get; set; }

26 │ }

27 │ [TokenUserCheckHTTP]

28 │ public IActionResult Index()

29 │ {

30 │ ViewData["ListData"] = new sysQuyenModel(dbcontext).GetData();

31 │ ViewData["bAdd"] = Utility.checkPermissionIDModel2((int)objToken.userid, (int)MenuID.sysQuyen, (int)Permissi...

32 │ ViewData["bEdit"] = Utility.checkPermissionIDModel2((int)objToken.userid, (int)MenuID.sysQuyen, (int)Permiss...

33 │ ViewData["bDelete"] = Utility.checkPermissionIDModel2((int)objToken.userid, (int)MenuID.sysQuyen, (int)Permi...

34 │ return View();

35 │ }

36 │ [HttpGet("add")]

37 │ [TokenUserCheckHTTP]

38 │ public IActionResult Add()

39 │ {

40 │ ViewData["bAdd"] = Utility.checkPermissionIDModel2((int)objToken.userid, (int)MenuID.sysQuyen, (int)Permissi...

64 │ // when Result = 0 then failure

65 │ // when Result = -2 then not exist

66 │ // when Result = -3 then object is duplicate

67 │ public int Edit(int ID, string Code, int? ParentID, string Name, int Sequence, string Link, string Icon, bool St...

68 │ , int? UserID, int? ChildOfMenu, string UsingFor)

69 │ {

70 │ if (CheckExists(ID, Code))

71 │ return -3;

72 │

73 │ sys\_Menu menuUpdate = GetItem(ID);

74 │

75 │ if (menuUpdate != null)

76 │ {

77 │ menuUpdate.Code = Code;

78 │ menuUpdate.MenuCha = ParentID;

79 │ menuUpdate.Ten = Name;

80 │ menuUpdate.SoThuTu = Sequence;

81 │ menuUpdate.DuongDan = Link;

82 │ menuUpdate.Icon = Icon;

83 │ menuUpdate.MenuCon = ChildOfMenu;

84 │ menuUpdate.TrangThai = Status;

85 │ menuUpdate.ThamSo = UsingFor;

86 │ menuUpdate.NguoiSua = UserID;

87 │ menuUpdate.NgaySua = DateTime.Now;

88 │ return Update(menuUpdate);

89 │ }

90 │

91 │ return -2;

92 │ }

93 │

94 │ // Summary:

95 │ // Delete menu.

96 │ // Returns:

97 │ // Case when Result = 1 then successful

98 │ // when Result = 0 then failure

99 │ // when Result = -1 then cannot delete

100 │ // when Result = -2 then not exist

101 │ public int Delete(int ID, int UserID)

102 │ {

103 │ sys\_Menu drM = GetItem(ID);

104 │

105 │ if (drM != null)

106 │ {

107 │ //Check children

108 │ sys\_Menu? drPM = First(t => t.MenuCha == ID && !(t.DaXoa ?? false));

109 │ sys\_RoleMenu? drRM = \_dbContext.Set<sys\_RoleMenu>().FirstOrDefault(t => t.MaMenu == ID && !(t.DaXoa ?? f...

110 │

111 │ if (drPM != null || drRM != null)

112 │ return -1;

113 │ else

114 │ {

115 │ drM.DaXoa= true;

116 │ drM.NgayXoa = DateTime.Now;

117 │ drM.NguoiXoa = UserID;

118 │ return Update(drM);

119 │ }

120 │ }

121 │

122 │ return -2;

123 │ }

124 │ }

125 │ }

────────────────────────────────────────────────────────────

**📄 sys\_OTP.cs (CS, 17 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class sys\_OTP

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaUser { get; set; }

11 │

12 │ public string? OTP { get; set; }

13 │

14 │ public int? MaToken { get; set; }

15 │

16 │ public DateTime? CreateDate { get; set; }

17 │ }

────────────────────────────────────────────────────────────

**📄 sys\_Quyen.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class sys\_Quyen

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public string Ten { get; set; } = null!;

13 │

14 │ public string? MoTa { get; set; }

15 │

16 │ public bool? TruyCapForm { get; set; }

17 │

18 │ public bool? TrangThai { get; set; }

19 │

20 │ public int? NguoiTao { get; set; }

21 │

22 │ public DateTime? NgayTao { get; set; }

23 │

24 │ public int? NguoiSua { get; set; }

25 │

26 │ public DateTime? NgaySua { get; set; }

27 │

28 │ public bool? DaXoa { get; set; }

29 │

30 │ public int? NguoiXoa { get; set; }

31 │

32 │ public DateTime? NgayXoa { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 sys\_QuyenModel.cs (CS, 86 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyChatLuong.Common;

3 │ using System.Net;

4 │

5 │ namespace QuanLyChatLuong.Models

6 │ {

7 │ public class sysQuyenModel : Base<sys\_Quyen>

8 │ {

9 │ public sysQuyenModel(DbContext dbContext) : base(dbContext)

10 │ { }

11 │ public sys\_Quyen GetItem(int? permissionID)

12 │ {

13 │ return First(t => t.Ma == permissionID && !(t.DaXoa ?? false));

14 │ }

15 │

16 │ public List<sys\_Quyen> GetData()

17 │ {

18 │ return Get(t => !(t.DaXoa ?? false)).OrderByDescending(t => t.Ma).ToList();

19 │ }

20 │ public int Add(string Code, string Name, string Description, bool FormAccess, bool Status, int UserID)

21 │ {

22 │ if (CheckExists(Utility.defaultUID, Code))

23 │ return -3;

24 │

25 │ sys\_Quyen roleAdd = new sys\_Quyen

26 │ {

27 │ Code = Code,

28 │ Ten = Name,

29 │ MoTa= Description,

30 │ TruyCapForm = FormAccess,

31 │ TrangThai = Status,

32 │ NguoiTao = UserID,

33 │ NgayTao = DateTime.Now

34 │ };

35 │ return Add(roleAdd);

36 │ }

37 │ public int Edit(int ID, string Code, string Name, string Description, bool FormAccess, bool Status, int UserID)

38 │ {

39 │ if (CheckExists(ID, Code))

40 │ return -3;

41 │

42 │ sys\_Quyen roleUpdate = GetItem(ID);

43 │

44 │ if (roleUpdate != null)

45 │ {

46 │ roleUpdate.Code = Code;

47 │ roleUpdate.Ten = Name;

48 │ roleUpdate.MoTa = Description;

49 │ roleUpdate.TruyCapForm = FormAccess;

50 │ roleUpdate.TrangThai = Status;

51 │ roleUpdate.NguoiSua = UserID;

52 │ roleUpdate.NgaySua = DateTime.Now;

53 │ return Update(roleUpdate);

54 │ }

55 │

56 │ return -2;

57 │ }

58 │ public int Delete(int ID, int UserID)

59 │ {

60 │ sys\_Quyen drR = GetItem(ID);

61 │

62 │ if (drR != null)

63 │ {

64 │ //Check children

65 │ sys\_RoleQuyen? drRolePermission = \_dbContext.Set<sys\_RoleQuyen>().FirstOrDefault(t => t.MaQuyen == ID &&...

66 │

67 │ if (drRolePermission != null)

68 │ return -1;

69 │ else

70 │ {

71 │ drR.DaXoa = true;

72 │ drR.NgayXoa = DateTime.Now;

73 │ drR.NguoiXoa = UserID;

74 │ return Update(drR);

75 │ }

76 │ }

77 │

78 │ return -2;

79 │ }

80 │ private bool CheckExists(int ID, string Code)

81 │ {

82 │ return Any(t => t.Ma != ID && t.Code == Code && !(t.DaXoa ?? false));

83 │ }

84 │ }

85 │ }

86 │

────────────────────────────────────────────────────────────

**📄 sys\_Role.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class sys\_Role

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string Code { get; set; } = null!;

11 │

12 │ public string Ten { get; set; } = null!;

13 │

14 │ public string? MoTa { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 sys\_RoleMenu.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class sys\_RoleMenu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaRole { get; set; }

11 │

12 │ public int MaMenu { get; set; }

13 │

14 │ public bool? ChoPhepTruyCap { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 sys\_RoleMenuModel.cs (CS, 93 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyChatLuong.Common;

3 │

4 │ namespace QuanLyChatLuong.Models

5 │ {

6 │ public class sysRoleMenuModel : Base<sys\_RoleMenu>

7 │ {

8 │ public sysRoleMenuModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │

12 │ public List<sp\_RoleMenuResult> GetRoleMenuFollowRoleID(int RoleID)

13 │ {

14 │ return \_dbContext.Set<sp\_RoleMenuResult>().FromSqlRaw($"sp\_RoleMenu @iRole='{RoleID}'").ToList();

15 │ }

16 │ public sys\_RoleMenu GetItem(int RoleMenuID)

17 │ {

18 │ return First(t => t.Ma == RoleMenuID && !(t.DaXoa ?? false));

19 │ }

20 │ public List<sys\_RoleMenu> GetDataFollowRoleID(int RoleID)

21 │ {

22 │ return Get(t => t.MaRole == RoleID && !(t.DaXoa ?? false)).OrderByDescending(t => t.NgayTao).ToList();

23 │ }

24 │ public sys\_RoleMenu GetItem(int RoleID, int MenuID)

25 │ {

26 │ return First(t => t.MaRole == RoleID && t.MaMenu == MenuID && !(t.DaXoa ?? false));

27 │ }

28 │ public int Add(int RoleID, int MenuID, bool AllowAccess, bool Status, int UserID)

29 │ {

30 │ if (CheckExists(Utility.defaultUID, RoleID, MenuID))

31 │ return -3;

32 │

33 │ sys\_RoleMenu roleMenuAdd = new sys\_RoleMenu

34 │ {

35 │ MaRole = RoleID,

36 │ MaMenu = MenuID,

37 │ ChoPhepTruyCap = AllowAccess,

38 │ TrangThai = Status,

39 │ NguoiTao = UserID,

40 │ NgayTao = DateTime.Now,

41 │ };

42 │ return Add(roleMenuAdd);

43 │ }

44 │ public int Edit(int ID, int MenuID, bool AllowAccess, bool Status, int UserID)

45 │ {

46 │ sys\_RoleMenu roleMenuUpdate = GetItem(ID);

47 │

48 │ if (roleMenuUpdate != null)

49 │ {

50 │ if (CheckExists(ID, roleMenuUpdate.MaRole, MenuID))

51 │ return -3;

52 │

53 │ roleMenuUpdate.MaMenu = MenuID;

54 │ roleMenuUpdate.ChoPhepTruyCap = AllowAccess;

55 │ roleMenuUpdate.TrangThai = Status;

56 │ roleMenuUpdate.NguoiSua = UserID;

57 │ roleMenuUpdate.NgaySua = DateTime.Now;

58 │

59 │ return Update(roleMenuUpdate);

60 │ }

61 │

62 │ return -2;

63 │ }

64 │ public int Delete(int ID, int UserID)

65 │ {

66 │ sys\_RoleMenu drRM = GetItem(ID);

67 │

68 │ if (drRM != null)

69 │ {

70 │ //Check children

71 │ sys\_RoleQuyen? drRP = \_dbContext.Set<sys\_RoleQuyen>().FirstOrDefault(t => (t.MaRole == drRM.MaRole && t....

72 │

73 │ if (drRP != null)

74 │ return -1;

75 │ else

76 │ {

77 │ drRM.DaXoa = true;

78 │ drRM.NgayXoa = DateTime.Now;

79 │ drRM.NguoiXoa = UserID;

80 │

81 │ return Update(drRM);

82 │ }

83 │ }

84 │

85 │ return -2;

86 │ }

87 │ private bool CheckExists(int ID, int RoleID, int MenuID)

88 │ {

89 │ return Any(t => t.Ma != ID && t.MaRole == RoleID && t.MaMenu == MenuID && !(t.DaXoa ?? false));

90 │ }

91 │

92 │ }

93 │ }

────────────────────────────────────────────────────────────

**📄 sys\_RoleModel.cs (CS, 84 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyChatLuong.Common;

3 │

4 │ namespace QuanLyChatLuong.Models

5 │ {

6 │ public class sysRoleModel : Base<sys\_Role>

7 │ {

8 │ public sysRoleModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │ public List<sys\_Role> GetData()

12 │ {

13 │ return Get(t => !(t.DaXoa ?? false)).OrderByDescending(t => t.Ma).ToList();

14 │ }

15 │

16 │ public sys\_Role GetItem(int RoleID)

17 │ {

18 │ return First(t => t.Ma == RoleID && !(t.DaXoa ?? false));

19 │ }

20 │ public int Add(string Code, string Name, string Description, bool Status, int UserID)

21 │ {

22 │ if (CheckExists(Utility.defaultUID, Code))

23 │ return -3;

24 │

25 │ sys\_Role roleAdd = new sys\_Role

26 │ {

27 │ Code = Code,

28 │ Ten = Name,

29 │ MoTa = Description,

30 │ TrangThai = Status,

31 │ NguoiTao = UserID,

32 │ NgayTao = DateTime.Now

33 │ };

34 │ return Add(roleAdd);

35 │ }

36 │ public int Edit(int ID, string Code, string Name, string Description, bool Status, int UserID)

37 │ {

38 │ if (CheckExists(ID, Code))

39 │ return -3;

40 │

41 │ sys\_Role roleUpdate = GetItem(ID);

42 │

43 │ if (roleUpdate != null)

44 │ {

45 │ roleUpdate.Code = Code;

46 │ roleUpdate.Ten = Name;

47 │ roleUpdate.MoTa = Description;

48 │ roleUpdate.TrangThai = Status;

49 │ roleUpdate.NguoiSua = UserID;

50 │ roleUpdate.NgaySua = DateTime.Now;

51 │ return Update(roleUpdate);

52 │ }

53 │

54 │ return -2;

55 │ }

56 │ public int Delete(int ID, int UserID)

57 │ {

58 │ sys\_Role drR = GetItem(ID);

59 │

60 │ //if (drR != null)

61 │ //{

62 │ // //Check children

63 │ // SysRoleMenu? drRM = \_dbContext.Set<SysRoleMenu>().FirstOrDefault(t => t.RoleId == ID && !(t.Deleted ??...

64 │ // SysUserRole? drUR = \_dbContext.Set<SysUserRole>().FirstOrDefault(t => t.RoleId == ID && !(t.Deleted ??...

65 │

66 │ // if (drUR != null || drRM != null)

67 │ // return -1;

68 │ // else

69 │ // {

70 │ // drR.Deleted = true;

71 │ // drR.DeletedDate = DateTime.Now;

72 │ // drR.UserDeleted = UserID;

73 │ // return Update(drR);

74 │ // }

75 │ //}

76 │

77 │ return -2;

78 │ }

79 │ private bool CheckExists(int ID, string Code)

80 │ {

81 │ return Any(t => t.Ma != ID && t.Code == Code && !(t.DaXoa ?? false));

82 │ }

83 │ }

84 │ }

────────────────────────────────────────────────────────────

**📄 sys\_RolePermissionModel.cs (CS, 238 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyChatLuong.Common;

3 │

4 │ namespace QuanLyChatLuong.Models

5 │ {

6 │ public class sysRolePermissionModel : Base<sys\_RoleQuyen>

7 │ {

8 │ private class RoleMenu

9 │ {

10 │ public int Ma { get; set; }

11 │ public int MaMenu { get; set; }

12 │ public bool AllowAccess { get; set; }

13 │ public bool Status { get; set; }

14 │ }

15 │

16 │ private class RolePermission

17 │ {

18 │ public int Ma { get; set; }

19 │ public int MaMenu { get; set; }

20 │ public int MaQuyen { get; set; }

21 │ public bool Status { get; set; }

22 │ }

23 │ public sysRolePermissionModel(DbContext dbContext) : base(dbContext)

24 │ {

25 │ }

26 │ public sys\_RoleQuyen GetItem(int RolePermissionID)

27 │ {

28 │ return First(t => t.Ma == RolePermissionID && !(t.DaXoa ?? false));

29 │ }

30 │ public List<sys\_RoleQuyen> GetDataFollowRoleMenuID(int MenuID, int RoleID)

31 │ {

32 │ return Get(t => (t.MaRole == RoleID && t.MaMenu == MenuID) && !(t.DaXoa ?? false)).OrderByDescending(t => t....

33 │ }

34 │ public List<sp\_RolePermissionResult> GetRolePermissionFollowRoleID(int RoleID)

35 │ {

36 │ return \_dbContext.Set<sp\_RolePermissionResult>().FromSqlRaw($"exec sp\_RolePermission @iRole='{RoleID}'").ToL...

37 │ }

38 │

39 │ public int Delete(int ID, int UserID)

40 │ {

41 │ sys\_RoleQuyen drRP = GetItem(ID);

42 │ if (drRP != null)

43 │ {

44 │ drRP.DaXoa = true;

45 │ drRP.NgayXoa = DateTime.Now;

46 │ drRP.NguoiXoa = UserID;

47 │ return Update(drRP);

48 │ }

49 │ return -2;

50 │ }

51 │ public int Add(int RoleID, int MenuID, int PermissionID, bool Status, int UserID)

52 │ {

53 │ if (CheckExists(Utility.defaultUID, RoleID, MenuID, PermissionID))

54 │ return -3;

55 │

56 │ sys\_RoleQuyen rolePermissionAdd = new sys\_RoleQuyen

57 │ {

58 │ MaRole = RoleID,

59 │ MaMenu = MenuID,

60 │ MaQuyen = PermissionID,

61 │ TrangThai = Status,

62 │ NguoiTao = UserID,

63 │ NgayTao = DateTime.Now

64 │ };

65 │

66 │ return Add(rolePermissionAdd);

67 │ }

68 │

69 │ public int Edit(int ID, int PermissionID, bool Status, int UserID)

70 │ {

71 │ sys\_RoleQuyen rolePermissionUpdate = GetItem(ID);

72 │ if (rolePermissionUpdate != null)

73 │ {

74 │ rolePermissionUpdate.MaQuyen = PermissionID;

75 │ rolePermissionUpdate.TrangThai = Status;

76 │ rolePermissionUpdate.NguoiSua = UserID;

77 │ rolePermissionUpdate.NgaySua = DateTime.Now;

78 │ return Update(rolePermissionUpdate);

79 │ }

80 │ return -2;

81 │ }

82 │ public bool UpdateQuicklyPermission(int RoleID, string RMAllowAccess, string RMStatus, string RPStatus, int User...

83 │ {

84 │ //..

85 │ List<RoleMenu>? ListRoleMenu = new List<RoleMenu>();

86 │ List<RolePermission> ListRolePermisson = new List<RolePermission>();

87 │ string[]? ArrayItem = null;

88 │

89 │ //Xu ly RMAllowAccess

90 │ string[] Array = RMAllowAccess.Split(',');

91 │

92 │ if (Array != null && Array.Count() > 0)

93 │ {

94 │ for (int i = 0; i < Array.Count(); i++)

95 │ {

96 │ ArrayItem = Array[i].Split('.');

97 │

98 │ if (ArrayItem != null && ArrayItem.Count() == 2)

99 │ {

100 │ RoleMenu roleMenuAdd = new RoleMenu

101 │ {

102 │ Ma = Cast.ToInt(ArrayItem[0]),

103 │ MaMenu = Cast.ToInt(ArrayItem[1]),

104 │ AllowAccess = true,

105 │ Status = false

106 │ };

107 │ ListRoleMenu.Add(roleMenuAdd);

108 │ }

109 │ }

110 │ ListRoleMenu = ListRoleMenu.GroupBy(t => new { t.Ma, t.MaMenu }).Select(t => t.FirstOrDefault()).OrderBy...

111 │ }

112 │

113 │ //Xu ly RMStatus

114 │ Array = RMStatus.Split(',');

115 │ int itemRoleMenuID; int itemMenuID;

116 │ RoleMenu? itemRoleMenu = null;

117 │

118 │ if (Array != null && Array.Count() > 0)

119 │ {

120 │ for (int i = 0; i < Array.Count(); i++)

121 │ {

122 │ ArrayItem = Array[i].Split('.');

123 │

124 │ if (ArrayItem != null && ArrayItem.Count() == 2)

125 │ {

126 │ itemRoleMenuID = ArrayItem[0].GetGuid();

127 │ itemMenuID = ArrayItem[1].GetGuid();

128 │ itemRoleMenu = ListRoleMenu.FirstOrDefault(t => t.Ma == itemRoleMenuID && t.MaMenu == itemMenuID...

129 │

130 │ if (itemRoleMenu == null)

131 │ {

132 │ RoleMenu roleMenuAdd = new RoleMenu

133 │ {

134 │ Ma = itemRoleMenuID,

135 │ MaMenu = itemMenuID,

136 │ AllowAccess = false,

137 │ Status = true

138 │ };

139 │ ListRoleMenu.Add(roleMenuAdd);

140 │ }

141 │ else

142 │ itemRoleMenu.Status = true;

143 │ }

144 │ }

145 │ }

146 │

147 │ //Xu ly RPStatus

148 │ Array = RPStatus.Split(',');

149 │

150 │ if (Array != null && Array.Count() > 0)

151 │ {

152 │ for (int i = 0; i < Array.Count(); i++)

153 │ {

154 │ ArrayItem = Array[i].Split('.');

155 │

156 │ if (ArrayItem != null && ArrayItem.Count() == 3)

157 │ {

158 │ RolePermission rolePermissionAdd = new RolePermission

159 │ {

160 │ Ma = Cast.ToInt(ArrayItem[0]),

161 │ MaMenu = Cast.ToInt(ArrayItem[1]),

162 │ MaQuyen = Cast.ToInt(ArrayItem[2]),

163 │ Status = true

164 │ };

165 │ ListRolePermisson.Add(rolePermissionAdd);

166 │ }

167 │ }

168 │ }

169 │

170 │ //..

171 │ if (ListRoleMenu != null && ListRolePermisson != null)

172 │ {

173 │ //Update false role menu, permission exist

174 │ bool bUpdateRMP = UpdateRoleMenuPermission(RoleID);

175 │

176 │ if (bUpdateRMP)

177 │ {

178 │ int iCountUpdateRoleMenu = 0, iCountUpdateRolePermission = 0;

179 │

180 │ //Update Role Menu

181 │ sysRoleMenuModel sys\_RoleMenuM = new sysRoleMenuModel(\_dbContext);

182 │ foreach (RoleMenu item in ListRoleMenu)

183 │ {

184 │ if (item.Ma == Utility.defaultUID)//Add

185 │ iCountUpdateRoleMenu += sys\_RoleMenuM.Add(RoleID, item.MaMenu, item.AllowAccess, item.Status...

186 │ else//Edit

187 │ {

188 │ sys\_RoleMenu drRM = sys\_RoleMenuM.GetItem(item.Ma);

189 │

190 │ if (drRM != null)

191 │ iCountUpdateRoleMenu += sys\_RoleMenuM.Edit(drRM.Ma, drRM.MaMenu, item.AllowAccess, item....

192 │ }

193 │ }

194 │

195 │ //Update Role Permission

196 │ sysRolePermissionModel sys\_RolePermissionM = new sysRolePermissionModel(\_dbContext);

197 │ foreach (RolePermission item in ListRolePermisson)

198 │ {

199 │ if (item.Ma == Utility.defaultUID)//Add

200 │ {

201 │ sys\_RoleMenu drRM = sys\_RoleMenuM.GetItem(RoleID, item.MaMenu);

202 │

203 │ if (drRM != null)

204 │ iCountUpdateRolePermission += sys\_RolePermissionM.Add(drRM.MaRole,drRM.MaMenu, item.MaQu...

205 │ }

206 │ else//Edit

207 │ {

208 │ sys\_RoleQuyen drRP = sys\_RolePermissionM.GetItem(item.Ma);

209 │

210 │ if (drRP != null)

211 │ iCountUpdateRolePermission += sys\_RolePermissionM.Edit(drRP.Ma, drRP.MaQuyen, item.Statu...

212 │ }

213 │ }

214 │

215 │ //Verify

216 │ if (ListRoleMenu.Count == iCountUpdateRoleMenu && ListRolePermisson.Count == iCountUpdateRolePermiss...

217 │ return true;

218 │ return false;

219 │ }

220 │

221 │ return false;

222 │ }

223 │

224 │ return false;

225 │ }

226 │

227 │ private bool UpdateRoleMenuPermission(int RoleID)

228 │ {

229 │ if (\_dbContext.Database.ExecuteSqlRaw($"sp\_UpdateRoleMenuPermission @iRole='{RoleID}'") >= 0)

230 │ return true;

231 │ return false;

232 │ }

233 │ private bool CheckExists(int ID, int RoleID, int MenuID, int PermissionID)

234 │ {

235 │ return Any(t => t.Ma != ID && (t.MaRole == RoleID && t.MaMenu == MenuID) && t.MaQuyen == PermissionID && !(t...

236 │ }

237 │ }

238 │ }

────────────────────────────────────────────────────────────

**📄 sys\_RoleQuyen.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class sys\_RoleQuyen

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaRole { get; set; }

11 │

12 │ public int MaMenu { get; set; }

13 │

14 │ public int MaQuyen { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 sys\_User.cs (CS, 45 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class sys\_User

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string TenDangNhap { get; set; } = null!;

11 │

12 │ public string Password { get; set; } = null!;

13 │

14 │ public string HoTen { get; set; } = null!;

15 │

16 │ public bool? TuChoiNhanEmail { get; set; }

17 │

18 │ public bool? GioiTinh { get; set; }

19 │

20 │ public string? Email { get; set; }

21 │

22 │ public byte[]? Photo { get; set; }

23 │

24 │ public bool? TrangThai { get; set; }

25 │

26 │ public int? NguoiTao { get; set; }

27 │

28 │ public DateTime? NgayTao { get; set; }

29 │

30 │ public int? NguoiSua { get; set; }

31 │

32 │ public DateTime? NgaySua { get; set; }

33 │

34 │ public bool? DaXoa { get; set; }

35 │

36 │ public int? NguoiXoa { get; set; }

37 │

38 │ public DateTime? NgayXoa { get; set; }

39 │

40 │ public string? MSNV { get; set; }

41 │

42 │ public string? FCM { get; set; }

43 │

44 │ public string? SDT { get; set; }

45 │ }

────────────────────────────────────────────────────────────

**📄 sys\_UserModel.cs (CS, 161 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyChatLuong.Common;

3 │

4 │ namespace QuanLyChatLuong.Models

5 │ {

6 │ public class sysUserModel : Base<sys\_User>

7 │ {

8 │ public sysUserModel(DbContext dbContext) : base(dbContext)

9 │ {

10 │ }

11 │

12 │ public sys\_User GetItem(int? UserID)

13 │ {

14 │ return First(t => t.Ma == UserID && !(t.DaXoa ?? false));

15 │ }

16 │

17 │ public List<sys\_User> GetData()

18 │ {

19 │ return Get(t => !(t.DaXoa ?? false)).OrderByDescending(t => t.Ma).ToList();

20 │ }

21 │ private bool CheckExists(int ID, string Username)

22 │ {

23 │ return Any(t => t.Ma != ID && t.TenDangNhap == Username && !(t.DaXoa ?? false));

24 │ }

25 │ public int Edit(int ID, string Username, string FullName, int GioiTinh,

26 │ string MSNV, bool Status, int UserID)

27 │ {

28 │ if (CheckExists(ID, Username))

29 │ {

30 │ return -3;

31 │ }

32 │

33 │ sys\_User userUpdate = GetItem(ID);

34 │

35 │ if (userUpdate != null)

36 │ {

37 │ userUpdate.TenDangNhap = Username;

38 │ userUpdate.HoTen = FullName;

39 │ userUpdate.GioiTinh = GioiTinh == 1;

40 │ userUpdate.MSNV = MSNV;

41 │ userUpdate.TrangThai = Status;

42 │ userUpdate.NguoiSua = UserID;

43 │ userUpdate.NgaySua = DateTime.Now;

44 │

45 │ try

46 │ {

47 │ return Update(userUpdate);

48 │ }

49 │ catch (Exception ex)

50 │ {

51 │ string ss = ex.Message;

52 │ return 0;

53 │ }

54 │

55 │ }

56 │ return -2;

57 │ }

58 │ public (int, byte[]?) UpdateAvatar(int ID, byte[] HinhAnh)

59 │ {

60 │ sys\_User userUpdate = GetItem(ID);

61 │

62 │ if (userUpdate != null )

63 │ {

64 │ userUpdate.Photo = HinhAnh != null ? HinhAnh : userUpdate.Photo;

65 │ return (Update(userUpdate), userUpdate.Photo ?? null);

66 │ }

67 │ return (-2, null);

68 │ }

69 │

70 │ public int Add(string Username, string FullName, string Password, int GioiTinh, string MSNV, bool Status, int Us...

71 │ {

72 │ if (CheckExists(Utility.defaultUID, Username))

73 │ {

74 │ return -3;

75 │ }

76 │ sys\_User userAdd = new sys\_User

77 │ {

78 │ TenDangNhap = Username,

79 │ Password = Utility.GetSHA512(Password),

80 │ HoTen = FullName,

81 │ GioiTinh = GioiTinh == 0,

82 │ MSNV = MSNV,

83 │ TrangThai = Status,

84 │ NguoiTao = UserID,

85 │ NgayTao = DateTime.Now

86 │ };

87 │ return Add(userAdd);

88 │ }

89 │

90 │ public int Delete(int ID, int UserID)

91 │ {

92 │ sys\_User drU = GetItem(ID);

93 │

94 │ if (drU != null)

95 │ {

96 │ sys\_UserRole? drUR = \_dbContext.Set<sys\_UserRole>().FirstOrDefault(t => t.MaUser == ID && !(t.DaXoa ?? f...

97 │

98 │ if (drUR != null)

99 │ {

100 │ return -1;

101 │ }

102 │ else

103 │ {

104 │ drU.DaXoa = true;

105 │ drU.NgayXoa= DateTime.Now;

106 │ drU.NguoiXoa= UserID;

107 │ return Update(drU);

108 │ }

109 │ }

110 │

111 │ return -2;

112 │ }

113 │

114 │ public int ResetPassword(int ID, string Password)

115 │ {

116 │ sys\_User drU = GetItem(ID);

117 │ if (drU != null)

118 │ {

119 │ drU.Password = Utility.GetSHA512(Password);

120 │ return Update(drU);

121 │ }

122 │

123 │ return -1;

124 │ }

125 │

126 │ public int ChangePassword( int ID, string OldPassword, string NewPassword)

127 │ {

128 │ sys\_User drU = GetItem(ID);

129 │

130 │ if (drU != null)

131 │ {

132 │ if (drU.Password == Utility.GetSHA512(OldPassword))

133 │ return ResetPassword(ID, NewPassword);

134 │ return -2;

135 │ }

136 │

137 │ return -1;

138 │ }

139 │ public bool CheckUsername(string Username, int id)

140 │ {

141 │ return Any(x => x.TenDangNhap == Username && !(x.DaXoa ?? false) && (x.TrangThai ?? false) && (id == 0 || x....

142 │ }

143 │ public bool CheckMSNV(string Msnv, int id)

144 │ {

145 │ return Any(x => x.MSNV == Msnv && !(x.DaXoa ?? false) && (x.TrangThai ?? false) && (id == 0 || x.Ma!=id ));

146 │ }

147 │

148 │ public int UpdateFCM(int ID, string FCM)

149 │ {

150 │ sys\_User userUpdate = GetItem(ID);

151 │

152 │ if (userUpdate != null)

153 │ {

154 │ userUpdate.FCM = FCM;

155 │ int result = Update(userUpdate);

156 │ return result > 0 ? 1 : 0;

157 │ }

158 │ return -2;

159 │ }

160 │ }

161 │ }

────────────────────────────────────────────────────────────

**📄 sys\_UserRole.cs (CS, 31 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class sys\_UserRole

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaUser { get; set; }

11 │

12 │ public int MaRole { get; set; }

13 │

14 │ public int MaChiNhanh { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │

18 │ public int? NguoiTao { get; set; }

19 │

20 │ public DateTime? NgayTao { get; set; }

21 │

22 │ public int? NguoiSua { get; set; }

23 │

24 │ public DateTime? NgaySua { get; set; }

25 │

26 │ public bool? DaXoa { get; set; }

27 │

28 │ public int? NguoiXoa { get; set; }

29 │

30 │ public DateTime? NgayXoa { get; set; }

31 │ }

────────────────────────────────────────────────────────────

**📄 sys\_UserRoleModel.cs (CS, 79 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyChatLuong.Common;

3 │ using QuanLyChatLuong.Models;

4 │

5 │ namespace QuanLyChatLuong.Models

6 │ {

7 │ public class sysUserRoleModel : Base<sys\_UserRole>

8 │ {

9 │ public sysUserRoleModel(DbContext dbContext) : base(dbContext)

10 │ {

11 │ }

12 │

13 │ public sys\_UserRole GetItem(int UserRoleID)

14 │ {

15 │ return First(t => t.Ma == UserRoleID && !(t.DaXoa ?? false));

16 │ }

17 │

18 │ public List<sys\_UserRole> GetData()

19 │ {

20 │ return Get(t => !(t.DaXoa ?? false)).OrderByDescending(t => t.NgayTao).ToList();

21 │ }

22 │

23 │ public List<sys\_UserRole> GetDataFollowUserID(int UserID)

24 │ {

25 │ return Get(t => t.MaUser == UserID && !(t.DaXoa ?? false)).OrderByDescending(t => t.NgayTao).ToList();

26 │ }

27 │ public int Edit(int ID, int RoleID, bool Status, int UserID, int Branch)

28 │ {

29 │ sys\_UserRole userRoleUpdate = GetItem(ID);

30 │ if (userRoleUpdate != null)

31 │ {

32 │ if (CheckExists(ID, userRoleUpdate.MaUser, RoleID, Branch))

33 │ return -3;

34 │

35 │ userRoleUpdate.MaRole = RoleID;

36 │ userRoleUpdate.TrangThai = Status;

37 │ userRoleUpdate.NguoiSua = UserID;

38 │ userRoleUpdate.MaChiNhanh = Branch;

39 │ userRoleUpdate.NgaySua = DateTime.Now;

40 │ return Update(userRoleUpdate);

41 │ }

42 │

43 │ return -2;

44 │ }

45 │ public int Add(int UserID, int RoleID, bool Status, int UserCreatedID, int Branch)

46 │ {

47 │ if (CheckExists(Utility.defaultUID, UserID, RoleID, Branch))

48 │ return -3;

49 │

50 │ sys\_UserRole userRoleAdd = new sys\_UserRole

51 │ {

52 │ MaUser = UserID,

53 │ MaRole = RoleID,

54 │ TrangThai = Status,

55 │ MaChiNhanh = Branch,

56 │ NguoiTao = UserCreatedID,

57 │ NgayTao = DateTime.Now

58 │ };

59 │ return Add(userRoleAdd);

60 │ }

61 │ public int Delete(int ID, int UserID)

62 │ {

63 │ sys\_UserRole drUR = GetItem(ID);

64 │ if (drUR != null)

65 │ {

66 │ drUR.DaXoa = true;

67 │ drUR.NgayXoa = DateTime.Now;

68 │ drUR.NguoiXoa = UserID;

69 │ return Update(drUR);

70 │ }

71 │ return -2;

72 │ }

73 │

74 │ private bool CheckExists(int ID, int UserID, int RoleID, int MaChiNhanh)

75 │ {

76 │ return Any(t => t.Ma != ID && t.MaUser == UserID && t.MaRole == RoleID && t.MaChiNhanh == MaChiNhanh && !(t....

77 │ }

78 │ }

79 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_BanThanhPham.cs (CS, 87 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_BanThanhPham

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaSoLieu { get; set; }

11 │

12 │ public string? Gio { get; set; }

13 │

14 │ public string? Phut { get; set; }

15 │

16 │ public int? MaLo { get; set; }

17 │

18 │ public decimal? pH\_RO { get; set; }

19 │

20 │ public decimal? HCL { get; set; }

21 │

22 │ public decimal? NaOH { get; set; }

23 │

24 │ public decimal? HCL\_Loang { get; set; }

25 │

26 │ public decimal? NaOH\_Loang { get; set; }

27 │

28 │ public decimal? CQ\_SodiumCarbonat { get; set; }

29 │

30 │ public decimal? RasperChao { get; set; }

31 │

32 │ public string? HamLuongTBTrongBaMi { get; set; }

33 │

34 │ public string? pHSuaBot { get; set; }

35 │

36 │ public decimal? ND\_HoHoa { get; set; }

37 │

38 │ public decimal? DoNhot { get; set; }

39 │

40 │ public string? GhiChu { get; set; }

41 │

42 │ public string? NguoiPhanTich { get; set; }

43 │

44 │ public bool? DaXoa { get; set; }

45 │

46 │ public int? UserCreated { get; set; }

47 │

48 │ public DateTime? CreatedDate { get; set; }

49 │

50 │ public int? UserModified { get; set; }

51 │

52 │ public DateTime? ModifiedDate { get; set; }

53 │

54 │ public int? UserDeleted { get; set; }

55 │

56 │ public DateTime? DeletedDate { get; set; }

57 │

58 │ public int? MaBanThanhPham { get; set; }

59 │

60 │ public string? BauMeSuaBot { get; set; }

61 │

62 │ public decimal? ViscosityPeakBU { get; set; }

63 │

64 │ public decimal? ViscosityMax950C0MinBU { get; set; }

65 │

66 │ public decimal? ViscosityMax950C10MinBU { get; set; }

67 │

68 │ public decimal? ViscosityEndPointBU { get; set; }

69 │

70 │ public decimal? GelatinnizationTemperature { get; set; }

71 │

72 │ public decimal? SO2 { get; set; }

73 │

74 │ public decimal? AcetylGroup { get; set; }

75 │

76 │ public decimal? DS { get; set; }

77 │

78 │ public decimal? Clo { get; set; }

79 │

80 │ public string? Note { get; set; }

81 │

82 │ public string? NaHCO3 { get; set; }

83 │

84 │ public string? TinhBotTrongBaMi { get; set; }

85 │

86 │ public string? CamQuanNaHCO3 { get; set; }

87 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_ChiNhanh.cs (CS, 45 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_ChiNhanh

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string Ten { get; set; } = null!;

13 │

14 │ public string? TenDayDu { get; set; }

15 │

16 │ public string? DiaChi { get; set; }

17 │

18 │ public string? DienThoai { get; set; }

19 │

20 │ public string? Fax { get; set; }

21 │

22 │ public string? Website { get; set; }

23 │

24 │ public string? Email { get; set; }

25 │

26 │ public byte[]? Logo { get; set; }

27 │

28 │ public bool? TrangThai { get; set; }

29 │

30 │ public int? NguoiTao { get; set; }

31 │

32 │ public DateTime? NgayTao { get; set; }

33 │

34 │ public int? NguoiSua { get; set; }

35 │

36 │ public DateTime? NgaySua { get; set; }

37 │

38 │ public bool? DaXoa { get; set; }

39 │

40 │ public int? NguoiXoa { get; set; }

41 │

42 │ public DateTime? NgayXoa { get; set; }

43 │

44 │ public int? nID { get; set; }

45 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_ChiTietKiemTra.cs (CS, 41 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_ChiTietKiemTra

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaKTCL { get; set; }

11 │

12 │ public int? STT { get; set; }

13 │

14 │ public string? Code { get; set; }

15 │

16 │ public string? DVT { get; set; }

17 │

18 │ public string? TieuChuan { get; set; }

19 │

20 │ public string? KetQua { get; set; }

21 │

22 │ public bool? TrangThai { get; set; }

23 │

24 │ public int? NguoiTao { get; set; }

25 │

26 │ public DateTime? NgayTao { get; set; }

27 │

28 │ public int? NguoiSua { get; set; }

29 │

30 │ public DateTime? NgaySua { get; set; }

31 │

32 │ public bool? DaXoa { get; set; }

33 │

34 │ public int? NguoiXoa { get; set; }

35 │

36 │ public DateTime? NgayXoa { get; set; }

37 │

38 │ public string? Method { get; set; }

39 │

40 │ public int? MaTieuChuan { get; set; }

41 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_ChiTietSoLieu.cs (CS, 73 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_ChiTietSoLieu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaSoLieu { get; set; }

11 │

12 │ public string? Gio { get; set; }

13 │

14 │ public string? Phut { get; set; }

15 │

16 │ public decimal? B\_DoTrang { get; set; }

17 │

18 │ public decimal? B\_DoAm { get; set; }

19 │

20 │ public decimal? B\_DoMin { get; set; }

21 │

22 │ public decimal? B\_DoPH { get; set; }

23 │

24 │ public decimal? B\_DoKeoCP { get; set; }

25 │

26 │ public decimal? B\_DoKeoBU { get; set; }

27 │

28 │ public decimal? B\_TapChat { get; set; }

29 │

30 │ public decimal? ND\_Lo { get; set; }

31 │

32 │ public decimal? ND\_Say { get; set; }

33 │

34 │ public decimal? ND\_HoHoa { get; set; }

35 │

36 │ public decimal? NS\_DoPH { get; set; }

37 │

38 │ public decimal? BU\_DoAm { get; set; }

39 │

40 │ public string? Cepa { get; set; }

41 │

42 │ public decimal? XMS\_DoAm { get; set; }

43 │

44 │ public string? XMT\_CoBot { get; set; }

45 │

46 │ public decimal? HamLuongTB { get; set; }

47 │

48 │ public string? GhiChu { get; set; }

49 │

50 │ public int? NguoiTao { get; set; }

51 │

52 │ public DateTime? NgayTao { get; set; }

53 │

54 │ public int? NguoiSua { get; set; }

55 │

56 │ public DateTime? NgaySua { get; set; }

57 │

58 │ public bool? DaXoa { get; set; }

59 │

60 │ public int? NguoiXoa { get; set; }

61 │

62 │ public DateTime? NgayXoa { get; set; }

63 │

64 │ public int? MaLoaiBot { get; set; }

65 │

66 │ public int? MaLoThuMua { get; set; }

67 │

68 │ public decimal? SO2 { get; set; }

69 │

70 │ public string? ForeignMatter { get; set; }

71 │

72 │ public string? Sensory { get; set; }

73 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_ChiTieuCL.cs (CS, 15 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_ChiTieuCL

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public int? SoTT { get; set; }

15 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_ColorGioiHanSoLieu.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_ColorGioiHanSoLieu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int? MaChiNhanh { get; set; }

11 │

12 │ public string? Code { get; set; }

13 │

14 │ public string? MinColor { get; set; }

15 │

16 │ public string? MaxColor { get; set; }

17 │

18 │ public int? TrangThai { get; set; }

19 │

20 │ public int? NguoiTao { get; set; }

21 │

22 │ public DateTime? NgayTao { get; set; }

23 │

24 │ public int? NguoiSua { get; set; }

25 │

26 │ public DateTime? NgaySua { get; set; }

27 │

28 │ public bool? DaXoa { get; set; }

29 │

30 │ public int? NguoiXoa { get; set; }

31 │

32 │ public DateTime? NgayXoa { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_DanhMucSoLieu.cs (CS, 33 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_DanhMucSoLieu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public int? MaNhom { get; set; }

15 │

16 │ public int? MaLoai { get; set; }

17 │

18 │ public bool? TrangThai { get; set; }

19 │

20 │ public int? NguoiTao { get; set; }

21 │

22 │ public DateTime? NgayTao { get; set; }

23 │

24 │ public int? NguoiSua { get; set; }

25 │

26 │ public DateTime? NgaySua { get; set; }

27 │

28 │ public bool? DaXoa { get; set; }

29 │

30 │ public int? NguoiXoa { get; set; }

31 │

32 │ public DateTime? NgayXoa { get; set; }

33 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_GioiHanSoLieu.cs (CS, 45 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_GioiHanSoLieu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanh { get; set; }

11 │

12 │ public string Code { get; set; } = null!;

13 │

14 │ public string? Ten { get; set; }

15 │

16 │ public string? TenNgan { get; set; }

17 │

18 │ public string? Ten\_ZH { get; set; }

19 │

20 │ public string? Loai { get; set; }

21 │

22 │ public decimal? GiaTri { get; set; }

23 │

24 │ public string? DVT { get; set; }

25 │

26 │ public string? TieuChuan { get; set; }

27 │

28 │ public string? GhiChu { get; set; }

29 │

30 │ public bool? TrangThai { get; set; }

31 │

32 │ public int? NguoiTao { get; set; }

33 │

34 │ public DateTime? NgayTao { get; set; }

35 │

36 │ public int? NguoiSua { get; set; }

37 │

38 │ public DateTime? NgaySua { get; set; }

39 │

40 │ public bool? DaXoa { get; set; }

41 │

42 │ public int? NguoiXoa { get; set; }

43 │

44 │ public DateTime? NgayXoa { get; set; }

45 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_KiemTraChatLuong.cs (CS, 57 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_KiemTraChatLuong

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaCN { get; set; }

11 │

12 │ public string? TenSanPham { get; set; }

13 │

14 │ public string? SoDonHang { get; set; }

15 │

16 │ public DateTime? NgayGiaoHang { get; set; }

17 │

18 │ public string? SoLuong { get; set; }

19 │

20 │ public string? TenKhachHang { get; set; }

21 │

22 │ public DateTime? NgaySanXuat { get; set; }

23 │

24 │ public string? HanSuDung { get; set; }

25 │

26 │ public string? ChatLuongBot { get; set; }

27 │

28 │ public string? KetLuan { get; set; }

29 │

30 │ public string? GhiChu { get; set; }

31 │

32 │ public string? NguoiKiemTra { get; set; }

33 │

34 │ public bool? TrangThai { get; set; }

35 │

36 │ public int? NguoiTao { get; set; }

37 │

38 │ public DateTime? NgayTao { get; set; }

39 │

40 │ public int? NguoiSua { get; set; }

41 │

42 │ public DateTime? NgaySua { get; set; }

43 │

44 │ public bool? DaXoa { get; set; }

45 │

46 │ public int? NguoiXoa { get; set; }

47 │

48 │ public DateTime? NgayXoa { get; set; }

49 │

50 │ public string? MaLo { get; set; }

51 │

52 │ public string? VatLieuBaoBi { get; set; }

53 │

54 │ public string? TieuChuan { get; set; }

55 │

56 │ public int? KichThuoc { get; set; }

57 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_LoaiSoLieu.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_LoaiSoLieu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? TrangThai { get; set; }

15 │

16 │ public int? NguoiTao { get; set; }

17 │

18 │ public DateTime? NgayTao { get; set; }

19 │

20 │ public int? NguoiSua { get; set; }

21 │

22 │ public DateTime? NgaySua { get; set; }

23 │

24 │ public bool? DaXoa { get; set; }

25 │

26 │ public int? NguoiXoa { get; set; }

27 │

28 │ public DateTime? NgayXoa { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_ManLed.cs (CS, 17 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_ManLed

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? ipLed { get; set; }

11 │

12 │ public string? http { get; set; }

13 │

14 │ public int? MaCN { get; set; }

15 │

16 │ public bool? TrangThai { get; set; }

17 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_ManLedBusiness.cs (CS, 16 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │

3 │ namespace QuanLyChatLuong.Models

4 │ {

5 │ public class tbl\_ManLedBusiness : Base<tbl\_ManLed>

6 │ {

7 │ public tbl\_ManLedBusiness(DbContext dbContext) : base(dbContext)

8 │ {

9 │ }

10 │

11 │ public List<tbl\_ManLed> GetDataManLed(int MaCN)

12 │ {

13 │ return Get(t => t.MaCN == MaCN && (t.TrangThai ?? false)).OrderByDescending(t => t.Ma).ToList();

14 │ }

15 │ }

16 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_NhomSoLieu.cs (CS, 29 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_NhomSoLieu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public string? Code { get; set; }

11 │

12 │ public string? Ten { get; set; }

13 │

14 │ public bool? TrangThai { get; set; }

15 │

16 │ public int? NguoiTao { get; set; }

17 │

18 │ public DateTime? NgayTao { get; set; }

19 │

20 │ public int? NguoiSua { get; set; }

21 │

22 │ public DateTime? NgaySua { get; set; }

23 │

24 │ public bool? DaXoa { get; set; }

25 │

26 │ public int? NguoiXoa { get; set; }

27 │

28 │ public DateTime? NgayXoa { get; set; }

29 │ }

────────────────────────────────────────────────────────────

**📄 tbl\_SoLieu.cs (CS, 39 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tbl\_SoLieu

7 │ {

8 │ public int Ma { get; set; }

9 │

10 │ public int MaChiNhanh { get; set; }

11 │

12 │ public DateTime Ngay { get; set; }

13 │

14 │ public string? TruongCa { get; set; }

15 │

16 │ public string? QC { get; set; }

17 │

18 │ public string? ThuKhoBot { get; set; }

19 │

20 │ public string? TruongBPTP { get; set; }

21 │

22 │ public string? DotLo { get; set; }

23 │

24 │ public bool? TrangThai { get; set; }

25 │

26 │ public int? NguoiTao { get; set; }

27 │

28 │ public DateTime? NgayTao { get; set; }

29 │

30 │ public int? NguoiSua { get; set; }

31 │

32 │ public DateTime? NgaySua { get; set; }

33 │

34 │ public bool? DaXoa { get; set; }

35 │

36 │ public int? NguoiXoa { get; set; }

37 │

38 │ public DateTime? NgayXoa { get; set; }

39 │ }

────────────────────────────────────────────────────────────

**📄 tmp.cs (CS, 17 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tmp

7 │ {

8 │ public string? name { get; set; }

9 │

10 │ public decimal? \_12\_30\_PM\_\_5\_11 { get; set; }

11 │

12 │ public decimal? \_13\_30\_PM\_\_5\_11 { get; set; }

13 │

14 │ public decimal? \_14\_00\_PM\_\_5\_11 { get; set; }

15 │

16 │ public decimal? \_14\_30\_PM\_\_5\_11 { get; set; }

17 │ }

────────────────────────────────────────────────────────────

**📄 tmpDatum.cs (CS, 15 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class tmpDatum

7 │ {

8 │ public string? ThoiGian { get; set; }

9 │

10 │ public decimal B\_DoTrang { get; set; }

11 │

12 │ public decimal B\_DoAm { get; set; }

13 │

14 │ public decimal B\_TapChat { get; set; }

15 │ }

────────────────────────────────────────────────────────────

**📄 vPermissionModel.cs (CS, 17 lines)**

1 │ ﻿using Microsoft.EntityFrameworkCore;

2 │ using QuanLyChatLuong.Models.CRMEntity;

3 │ namespace QuanLyChatLuong.Models

4 │ {

5 │ public class vPermissionModel : Base<vQuyen>

6 │ {

7 │ public vPermissionModel(DbContext dbContext) : base(dbContext)

8 │ {

9 │ // disableCache = true;

10 │ }

11 │

12 │ public List<vQuyen> GetPermissionOfUser(int UserID, int BranchID)

13 │ {

14 │ return Get(t => t.MaUser == UserID && t.MaChiNhanh == BranchID).ToList();

15 │ }

16 │ }

17 │ }

────────────────────────────────────────────────────────────

**📄 vQuyen.cs (CS, 27 lines)**

1 │ ﻿using System;

2 │ using System.Collections.Generic;

3 │

4 │ namespace QuanLyChatLuong.Models;

5 │

6 │ public partial class vQuyen

7 │ {

8 │ public int MaRole { get; set; }

9 │

10 │ public int MaMenu { get; set; }

11 │

12 │ public int? MaQuyen { get; set; }

13 │

14 │ public int MaUser { get; set; }

15 │

16 │ public string TenDangNhap { get; set; } = null!;

17 │

18 │ public int MaChiNhanh { get; set; }

19 │

20 │ public string? Email { get; set; }

21 │

22 │ public bool TuChoiNhanEmail { get; set; }

23 │

24 │ public string HoTen { get; set; } = null!;

25 │ }

26 │

27 │