**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл |  |

СОГЛАСОВАНО  
Кандидат технических наук, приглашенный преподаватель департамента программной инженерии факультета компьютерных наук  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_ О. В. Максименкова  
«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. УТВЕРЖДАЮ  
Академический руководитель образовательной программы «Программная инженерия»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  
«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**Веб-приложение для волонтеров с элементами геймификации**

**Текст программы**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.01-01 12 01-1-ЛУ**

**Исполнитель**

Студент группы БПИ173

\_\_\_\_\_\_\_\_\_\_\_\_ / Королев Д. П. /

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**Москва 2020**

**УТВЕРЖДЕН**

**RU.17701729.04.01-01 12 01-1-ЛУ**

**Веб-приложение для волонтеров с элементами геймификации**

**Текст программы**

**RU.17701729.04.01-01 12 01-1**

**Листов 21**

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл |  |

**Москва 2020**

**СОДЕРЖАНИЕ**

1. [**ТЕКСТ ИСХОДНЫХ ФАЙЛОВ Volunteers** 4](#_Toc36947847)

[1.1. Файл Volunteers.Web.cs 4](#_Toc36947848)

[113.1. Файл Volunteers.Web.Program.cs 6](#_Toc36947849)

[197.1. Файл Volunteers.Web.Pages.Index.cshtml 7](#_Toc36947850)

[338.1. Файл Volunteers.Web.Pages.Index.cs 10](#_Toc36947851)

[363.1. Файл Volunteers.Web.Models.CultureTemplateRouteModelConvention cs 11](#_Toc36947852)

[393.1. Файл Volunteers.DataAccessLayer.Services.UserDbService.cs 11](#_Toc36947853)

[474.1. Файл Volunteers.DataAccessLayer.Services.ParticipationTokenDbService.cs 13](#_Toc36947854)

[589.1. Файл Volunteers.DataAccessLayer.Services.DbBaseService.cs 15](#_Toc36947855)

[655.1. Файл Volunteers.Core.CodeGenerator.cs 16](#_Toc36947856)

[695.1. Файл Volunteers.Common.Organization.cs 17](#_Toc36947857)

[734.1. Файл Volunteers.Common.ParticipationToken.cs 18](#_Toc36947858)

[734.2. Файл Volunteers.Common.SeedData.cs 18](#_Toc36947859)

[771.1. Файл Volunteers.Common.Helpers.IEnumHelper.cs 19](#_Toc36947860)

[791.1. Файл Volunteers.Common.Models.User.cs 19](#_Toc36947861)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 21](#_Toc36947862)

# **ТЕКСТ ИСХОДНЫХ ФАЙЛОВ Volunteers**

## Файл Volunteers.Web.cs

1. using System.Globalization;
2. using Microsoft.AspNetCore.Builder;
3. using Microsoft.AspNetCore.Identity;
4. using Microsoft.AspNetCore.Hosting;
5. using Microsoft.AspNetCore.Http;
6. using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
7. using Microsoft.AspNetCore.Identity.UI.Services;
8. using Microsoft.AspNetCore.Localization;
9. using Microsoft.AspNetCore.Mvc;
10. using Microsoft.AspNetCore.Mvc.ApplicationModels;
11. using Microsoft.AspNetCore.Routing;
12. using Microsoft.EntityFrameworkCore;
13. using Microsoft.Extensions.Configuration;
14. using Microsoft.Extensions.DependencyInjection;
15. using Microsoft.Extensions.Hosting;
16. using Microsoft.VisualBasic;
17. using Volunteers.Common.Models;
18. using Volunteers.Core.Models;
19. using Volunteers.DataAccessLibrary;
20. using Volunteers.DataAccessLibrary.Services;
21. using Volunteers.Web.Models;
22. namespace Volunteers.Web
23. {
24. public class Startup
25. {
26. public Startup(IConfiguration configuration)
27. {
28. Configuration = configuration;
29. }
30. public IConfiguration Configuration { get; }
31. // This method gets called by the runtime. Use this method to add services to the container.
32. public void ConfigureServices(IServiceCollection services)
33. {
34. // database context
35. services.AddDbContext<AppDbContext>(options =>
36. options.UseSqlServer(
37. Configuration.GetConnectionString("DefaultConnection")));
38. // identity setup
39. services.AddDefaultIdentity<User>(options => options.SignIn.RequireConfirmedAccount = true)
40. .AddRoles<IdentityRole>()
41. .AddEntityFrameworkStores<AppDbContext>();
42. services.AddScoped<UserManager<User>>();
43. // resolve identityDbContext as application db context
44. services.AddScoped<IdentityDbContext<User>, AppDbContext>();
45. // core services
46. services.AddScoped<TokenGenerator>();
47. services.AddScoped<CodeGenerator>();
48. services.AddScoped<LevelCounter>();
49. services.AddTransient<IEmailSender, EmailSender>();
50. services.Configure<AuthMessageSenderOptions>(Configuration);
51. // Database services
52. services.AddScoped<UserDbService>();
53. services.AddScoped<OrganizationDbService>();
54. services.AddScoped<EventDbService>();
55. services.AddScoped<ParticipationDbService>();
56. services.AddScoped<BonusDbService>();
57. // localization and pages
58. services.Configure<CookiePolicyOptions>(options =>
59. {
60. // This lambda determines whether user consent for non-essential cookies is needed for a given request.
61. options.CheckConsentNeeded = context => true;
62. options.MinimumSameSitePolicy = SameSiteMode.None;
63. });
64. services.ConfigureRequestLocalization();
65. services
66. .AddMvc(options =>
67. {
68. options.EnableEndpointRouting = false;
69. })
70. .AddViewLocalization(o => o.ResourcesPath = "Resources")
71. //.AddModelBindingMessagesLocalizer(services)
72. .AddRazorPagesOptions(options =>
73. {
74. options.Conventions.Add(new CultureTemplateRouteModelConvention());
75. });
76. }
77. // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
78. public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
79. {
80. if (env.IsDevelopment())
81. {
82. app.UseDeveloperExceptionPage();
83. }
84. else
85. {
86. app.UseExceptionHandler("/Error");
87. app.UseHsts();
88. }
89. app.UseRequestLocalization();
90. app.UseHttpsRedirection();
91. app.UseStaticFiles();
92. app.UseCookiePolicy();
93. app.UseAuthentication();
94. app.UseAuthorization();
95. app.UseMvc();
96. }
97. }
98. }

## Файл Volunteers.Web.Program.cs

1. using System;
2. using System.Collections.Generic;
3. using System.IO;
4. using System.Linq;
5. using System.Threading.Tasks;
6. using Microsoft.AspNetCore.Hosting;
7. using Microsoft.AspNetCore.Identity;
8. using Microsoft.Extensions.Configuration;
9. using Microsoft.Extensions.DependencyInjection;
10. using Microsoft.Extensions.Hosting;
11. using Microsoft.Extensions.Logging;
12. using Serilog;
13. using Volunteers.Common.Models;
14. using Volunteers.Core.Models;
15. using Volunteers.DataAccessLibrary;
16. using Volunteers.DataAccessLibrary.Models;
17. using Volunteers.DataAccessLibrary.Services;
18. namespace Volunteers.Web
19. {
20. public class Program
21. {
22. private static string \_environmentName;
23. public static async Task Main(string[] args)
24. {
25. var host = CreateHostBuilder(args).Build();
26. var configuration = new ConfigurationBuilder()
27. .SetBasePath(Directory.GetCurrentDirectory())
28. .AddJsonFile("appsettings.json")
29. .AddJsonFile($"appsettings.{\_environmentName}.json", optional: true, reloadOnChange: true)
30. .Build();
31. Log.Logger = new LoggerConfiguration()
32. .ReadFrom.Configuration(configuration)
33. .CreateLogger();
34. // before running host initialization
35. using (var scope = host.Services.CreateScope())
36. {
37. var services = scope.ServiceProvider;
38. try
39. {
40. var context = services.GetRequiredService<AppDbContext>();
41. var userManager = services.GetRequiredService<UserManager<User>>();
42. var roleManager = services.GetRequiredService<RoleManager<IdentityRole>>();
43. var organizationDbService = services.GetRequiredService<OrganizationDbService>();
44. var dbSeeder = new DbSeeder(roleManager,
45. userManager,
46. context,
47. organizationDbService);
48. await dbSeeder.SeedRoles();
49. await dbSeeder.SeedUsers();
50. await dbSeeder.ApplyRolesToUsers();
51. await dbSeeder.SeedOrganizations();
52. }
53. catch (Exception ex)
54. {
55. Log.Error(ex, "An error occurred while seeding the database.");
56. }
57. }
58. Log.Information("Launching application...");
59. await host.RunAsync();
60. }
61. public static IHostBuilder CreateHostBuilder(string[] args) =>
62. Host.CreateDefaultBuilder(args)
63. .ConfigureLogging((hostingContext, config) =>
64. {
65. config.ClearProviders();
66. \_environmentName = hostingContext.HostingEnvironment.EnvironmentName;
67. })
68. .ConfigureWebHostDefaults(webBuilder =>
69. {
70. webBuilder.UseStartup<Startup>();
71. });
72. }
73. }

## Файл Volunteers.Web.Pages.Index.cshtml

1. @page
2. @using System.Globalization
3. @using Microsoft.AspNetCore.Mvc.Localization
4. @inject IViewLocalizer Localizer
5. @model IndexModel
6. @{
7. ViewData["Title"] = "Home page";
8. }
9. <div class="text-center">
10. <div id="myCarousel" class="carousel slide" data-ride="carousel" wfd-id="17">
11. <ol class="carousel-indicators">
12. <li data-target="#myCarousel" data-slide-to="0" class="" wfd-id="30"></li>
13. <li data-target="#myCarousel" data-slide-to="1" wfd-id="29" class="active"></li>
14. <li data-target="#myCarousel" data-slide-to="2" wfd-id="28" class=""></li>
15. </ol>
16. <div class="carousel-inner" wfd-id="22">
17. <div class="carousel-item" wfd-id="25">
18. <svg width="100%" height="100%"
19. xmlns="http://www.w3.org/2000/svg"
20. preserveAspectRatio="xMidYMid slice"
21. focusable="false" role="img">
22. <linearGradient id="linear-gradient-left">
23. <stop offset="0%" stop-color="black" />
24. <stop offset="100%" stop-color="white" />
25. </linearGradient>
26. <rect fill="url(#linear-gradient-left)"
27. width="100%" height="100%" />
28. </svg>
29. <div class="container" wfd-id="26">
30. <div class="carousel-caption text-left" wfd-id="27">
31. <h1 class="text-white">Volunteers</h1>
32. <a class="text-white" href="#div\_takepartinevents">
33. <p>@Localizer["menu\_TakePartInEvents"]</p>
34. </a>
35. </div>
36. </div>
37. </div>
38. <div class="carousel-item active" wfd-id="24">
39. <svg width="100%" height="100%"
40. xmlns="http://www.w3.org/2000/svg"
41. preserveAspectRatio="xMidYMid slice"
42. focusable="false" role="img">
43. <linearGradient id="linear-gradient">
44. <stop offset="0%" stop-color="white" />
45. <stop offset="50%" stop-color="black" />
46. <stop offset="100%" stop-color="white" />
47. </linearGradient>
48. <rect fill="url(#linear-gradient)"
49. width="100%" height="100%" />
50. </svg>
51. <div class="container" wfd-id="42">
52. <div class="carousel-caption" wfd-id="43">
53. <h1 class="text-white">Volunteers</h1>
54. <a class="text-white" href="#div\_helpsociety">
55. <p>@Localizer["menu\_HelpSociety"]</p>
56. </a>
57. </div>
58. </div>
59. </div>
60. <div class="carousel-item" wfd-id="23">
61. <svg width="100%" height="100%"
62. xmlns="http://www.w3.org/2000/svg"
63. preserveAspectRatio="xMidYMid slice"
64. focusable="false" role="img">
65. <linearGradient id="linear-gradient-right">
66. <stop offset="0%" stop-color="white" />
67. <stop offset="100%" stop-color="black" />
68. </linearGradient>
69. <rect fill="url(#linear-gradient-right)"
70. width="100%" height="100%" />
71. </svg>
72. <div class="container" wfd-id="44">
73. <div class="carousel-caption text-right" wfd-id="45">
74. <h1 class="text-white">Volunteers</h1>
75. <a class="text-white" href="#div\_getbonuses">
76. <p>@Localizer["menu\_GetBonuses"]</p>
77. </a>
78. </div>
79. </div>
80. </div>
81. </div>
82. <a class="carousel-control-prev" href="#myCarousel" role="button" data-slide="prev">
83. <span class="carousel-control-prev-icon" aria-hidden="true" wfd-id="21"></span>
84. <span class="sr-only" wfd-id="20">@Localizer["menu\_Previous"]</span>
85. </a>
86. <a class="carousel-control-next" href="#myCarousel" role="button" data-slide="next">
87. <span class="carousel-control-next-icon" aria-hidden="true" wfd-id="19"></span>
88. <span class="sr-only" wfd-id="18">@Localizer["menu\_Next"]</span>
89. </a>
90. </div>
91. <hr class="mb-5 mt-5" id="div\_takepartinevents" />
92. <div class="row featurette" wfd-id="9">
93. <div class="col-md-7" wfd-id="11">
94. <h2 class="featurette-heading">
95. @Localizer["header\_TakePartInEvents"]
96. <br /><span class="text-muted" wfd-id="12">@Localizer["subheader\_TakePartInEvents"]</span>
97. </h2>
98. <p class="lead">@Localizer["content\_TakePartInEvents"]</p>
99. </div>
100. <div class="col-md-5" wfd-id="10">
101. <img class="h-100 w-100" src='@Localizer["img\_volunteerEvent"]' />
102. </div>
103. </div>
104. <hr class="mb-5 mt-5" id="div\_helpsociety" />
105. <div class="row featurette" wfd-id="9">
106. <div class="col-md-5" wfd-id="10">
107. <img class="h-100 w-100" src='@Localizer["img\_volunteerHelp"]' />
108. </div>
109. <div class="col-md-7" wfd-id="11">
110. <h2 class="featurette-heading">
111. @Localizer["header\_HelpSociety"]
112. <br /><span class="text-muted" wfd-id="12"> @Localizer["subheader\_HelpSociety"]</span>
113. </h2>
114. <p class="lead">@Localizer["content\_HelpSociety"]</p>
115. </div>
116. </div>
117. <hr class="mb-5 mt-5" id="div\_getbonuses" />
118. <div class="row featurette" wfd-id="9">
119. <div class="col-md-7" wfd-id="11">
120. <h2 class="featurette-heading">
121. @Localizer["header\_GetBonuses"]
122. <br /><span class="text-muted" wfd-id="12">@Localizer["subheader\_GetBonuses"]</span>
123. </h2>
124. <p class="lead">@Localizer["content\_GetBonuses"]</p>
125. </div>
126. <div class="col-md-5" wfd-id="10">
127. <img class="h-100 w-100" src='@Localizer["img\_volunteerBonus"]' />
128. </div>
129. </div>
130. </div>

## Файл Volunteers.Web.Pages.Index.cs

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Threading.Tasks;
5. using Microsoft.AspNetCore.Mvc;
6. using Microsoft.AspNetCore.Mvc.RazorPages;
7. using Microsoft.Extensions.Logging;
8. namespace Volunteers.Web.Pages
9. {
10. public class IndexModel : PageModel
11. {
12. private readonly ILogger<IndexModel> \_logger;
13. public IndexModel(ILogger<IndexModel> logger)
14. {
15. \_logger = logger;
16. }
17. public void OnGet()
18. {
19. }
20. }
21. }

## Файл Volunteers.Web.Models.CultureTemplateRouteModelConvention cs

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Threading.Tasks;
5. using Microsoft.AspNetCore.Mvc.ApplicationModels;
6. namespace Volunteers.Web.Models
7. {
8. public class CultureTemplateRouteModelConvention : IPageRouteModelConvention
9. {
10. public void Apply(PageRouteModel model)
11. {
12. var selectorCount = model.Selectors.Count;
13. for (var i = 0; i < selectorCount; i++)
14. {
15. var selector = model.Selectors[i];
16. model.Selectors.Add(new SelectorModel
17. {
18. AttributeRouteModel = new AttributeRouteModel
19. {
20. Order = -1,
21. Template = AttributeRouteModel.CombineTemplates(
22. "{culture?}",
23. selector.AttributeRouteModel.Template),
24. }
25. });
26. }
27. }
28. }
29. }

## Файл Volunteers.DataAccessLayer.Services.UserDbService.cs

1. using System;
2. using System.Collections.Generic;
3. using System.Security.Claims;
4. using System.Text;
5. using System.Threading.Tasks;
6. using Microsoft.AspNetCore.Http;
7. using Microsoft.AspNetCore.Identity;
8. using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
9. using Microsoft.EntityFrameworkCore;
10. using Volunteers.Common.Helpers;
11. using Volunteers.Common.Models;
12. using Volunteers.DataAccessLibrary.Models;
13. namespace Volunteers.DataAccessLibrary.Services
14. {
15. public class UserDbService : DbBaseService<User, string>
16. {
17. private readonly UserManager<User> \_userManager;
18. public UserDbService(
19. IdentityDbContext<User> identityDbContext,
20. UserManager<User> userManager)
21. : base(identityDbContext)
22. {
23. \_userManager = userManager;
24. }
25. public async Task<User> GetCurrentUser(ClaimsPrincipal claims)
26. {
27. return await \_userManager.GetUserAsync(claims);
28. }
29. public override async Task<List<User>> GetAll()
30. {
31. return await \_userManager.Users.ToListAsync();
32. }
33. public async Task MinusRatePoints(string userId, decimal points)
34. {
35. var itemInDb = await FindById(userId);
36. if (itemInDb.IsNull())
37. {
38. return;
39. }
40. itemInDb.PointsRate -= points;
41. await UpdateContext();
42. }
43. public async Task AssignRatePoints(string userId, decimal points)
44. {
45. var itemInDb = await FindById(userId);
46. if (itemInDb.IsNull())
47. {
48. return;
49. }
50. itemInDb.PointsRate += points;
51. await UpdateContext();
52. }
53. public override async Task Update(User item)
54. {
55. var itemInDb = await FindById(item.Id);
56. if (itemInDb.IsNull())
57. {
58. return;
59. }
60. itemInDb.PhoneNumber = item.PhoneNumber;
61. itemInDb.PointsRate = item.PointsRate;
62. await UpdateContext();
63. }
64. }
65. }

## Файл Volunteers.DataAccessLayer.Services.ParticipationTokenDbService.cs

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6. using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
7. using Microsoft.EntityFrameworkCore;
8. using Volunteers.Common.Models;
9. using Volunteers.Core.Models;
10. using Volunteers.DataAccessLibrary.Models;
11. namespace Volunteers.DataAccessLibrary.Services
12. {
13. public class ParticipationDbService : DbBaseService<ParticipantToken, int>
14. {
15. private readonly TokenGenerator \_tokenGenerator;
16. public ParticipationDbService(IdentityDbContext<User> dbContext,
17. TokenGenerator tokenGenerator)
18. : base(dbContext)
19. {
20. \_tokenGenerator = tokenGenerator;
21. }
22. public async Task<ParticipantToken> GetParticipantTokenByTokenValue(string token)
23. {
24. return await DbContext.Set<ParticipantToken>()
25. .FirstOrDefaultAsync(t => t.Token == token);
26. }
27. public async Task<List<int>> GetParticipatedEventIds(string userId)
28. {
29. return await DbContext.Set<ParticipantToken>()
30. .Where(t => t.UserId == userId)
31. .Select(t => t.EventId)
32. .ToListAsync();
33. }
34. public async Task AddNewParticipant(int eventId, string userId)
35. {
36. // if exists -> do nothing
37. var participationToken = await GetTokenOfUserInEvent(userId, eventId);
38. if (participationToken != null)
39. {
40. return;
41. }
42. var participantToken = new ParticipantToken
43. {
44. EventId = eventId,
45. UserId = userId,
46. Token = \_tokenGenerator.GenerateParticipateToken()
47. };
48. await Add(participantToken);
49. }
50. public async Task<ParticipantToken> FindParticipationTokenByUserIdAndEventId(string userId, int eventId)
51. {
52. return await DbContext.Set<ParticipantToken>()
53. .Where(p => p.UserId == userId)
54. .FirstOrDefaultAsync(p => p.EventId == eventId);
55. }
56. public async Task<ParticipantToken> GetTokenOfUserInEvent(string userId, int eventId)
57. {
58. var participationToken = await FindParticipationTokenByUserIdAndEventId(userId, eventId);
59. if (participationToken == null)
60. {
61. return null;
62. }
63. return participationToken;
64. }
65. public async Task CancelParticipation(string userId, int eventId)
66. {
67. var participationToken = await FindParticipationTokenByUserIdAndEventId(userId, eventId);
68. if (participationToken == null)
69. {
70. return;
71. }
72. await Delete(participationToken.Id);
73. }
74. public async Task DeactivateAttendanceToken(int id)
75. {
76. var participationToken = await FindById(id);
77. if (participationToken == null)
78. {
79. return;
80. }
81. participationToken.IsActivated = false;
82. await UpdateContext();
83. }
84. public async Task ActivateAttendanceToken(int id)
85. {
86. var participationToken = await FindById(id);
87. if (participationToken == null)
88. {
89. return;
90. }
91. participationToken.IsActivated = true;
92. await UpdateContext();
93. }
94. }
95. }

## Файл Volunteers.DataAccessLayer.Services.DbBaseService.cs

1. using System;
2. using System.Collections.Generic;
3. using System.Text;
4. using System.Threading.Tasks;
5. using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
6. using Microsoft.EntityFrameworkCore;
7. using Volunteers.Common.Helpers;
8. using Volunteers.Common.Models;
9. using Volunteers.DataAccessLibrary.Interfaces;
10. namespace Volunteers.DataAccessLibrary.Models
11. {
12. public class DbBaseService<TType, TKey> : IDbService<TType, TKey>
13. where TType : class
14. {
15. protected readonly IdentityDbContext<User> DbContext;
16. public DbBaseService(IdentityDbContext<User> dbContext)
17. {
18. DbContext = dbContext;
19. }
20. public virtual async Task Add(TType item)
21. {
22. await DbContext.AddAsync(item);
23. await UpdateContext();
24. }
25. public virtual async Task AddRange(IList<TType> items)
26. {
27. await DbContext.AddRangeAsync(items);
28. await UpdateContext();
29. }
30. public virtual async Task<TType> FindById(TKey key)
31. {
32. return await DbContext.FindAsync<TType>(key);
33. }
34. public virtual async Task Delete(TKey key)
35. {
36. var item = await FindById(key);
37. if (!item.IsNull())
38. {
39. DbContext.Remove(item);
40. await UpdateContext();
41. }
42. }
43. public virtual Task Update(TType item)
44. {
45. throw new **NotImplementedException**();
46. }
47. public virtual async Task<List<TType>> GetAll()
48. {
49. return await DbContext.Set<TType>().ToListAsync();
50. }
51. protected async Task UpdateContext()
52. {
53. await DbContext.SaveChangesAsync();
54. }
55. }
56. }

## Файл Volunteers.Core.CodeGenerator.cs

1. using System;
2. using System.Collections.Generic;
3. using System.Drawing;
4. using System.Drawing.Imaging;
5. using System.IO;
6. using System.Security.Cryptography;
7. using System.Text;
8. using QRCoder;
9. namespace Volunteers.Core.Models
10. {
11. public class CodeGenerator
12. {
13. private readonly QRCodeGenerator \_qrCodeGenerator
14. = new QRCodeGenerator();
15. private readonly Random \_randomGenerator = new Random();
16. public Bitmap GetQrCodeBitmap(string token)
17. {
18. var qrCodeData = \_qrCodeGenerator.CreateQrCode(token, QRCodeGenerator.ECCLevel.Q);
19. var qrCode = new QRCode(qrCodeData);
20. return qrCode.GetGraphic(20);
21. }
22. public string GetQrCodeStringContentForViewData(string token)
23. {
24. var qrBitmap = GetQrCodeBitmap(token);
25. using var memoryStream = new MemoryStream();
26. qrBitmap.Save(memoryStream, ImageFormat.Png);
27. return $"data:image/png;base64,{Convert.ToBase64String(memoryStream.ToArray())}";
28. }
29. public int GenerateBonusCode()
30. {
31. return \_randomGenerator.Next(0, 10000);
32. }
33. }
34. }

## Файл Volunteers.Common.Organization.cs

1. using System;
2. using System.Collections.Generic;
3. using System.ComponentModel.DataAnnotations;
4. using System.ComponentModel.DataAnnotations.Schema;
5. using System.Text;
6. namespace Volunteers.Common.Models
7. {
8. public class Organization
9. {
10. [Key]
11. public int Id { get; set; }
12. [Required]
13. public string Name { get; set; }
15. [Required]
16. [DataType(DataType.PhoneNumber)]
17. public string Phone { get; set; }
19. [Required]
20. [DataType(DataType.EmailAddress)]
21. public string Email { get; set; }
23. [Required]
24. public string Address { get; set; }
25. public string Description { get; set; }
26. #region Navigation Properties
27. public string UserId { get; set; }
28. public User Manager { get; set; }
29. public List<Event> Events { get; set; }
30. #endregion
31. }
32. }

## Файл Volunteers.Common.ParticipationToken.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Text;

namespace Volunteers.Common.Models

{

public class ParticipantToken

{

[Key]

public int Id { get; set; }

[ForeignKey("UserId")]

public string UserId { get; set; }

[ForeignKey("EventId")]

public int EventId { get; set; }

public string Token { get; set; }

public bool IsActivated { get; set; }

}

}

## Файл Volunteers.Common.SeedData.cs

1. using System;
2. using System.Collections.Generic;
3. using System.Text;
4. using Volunteers.Common.Models;
5. namespace Volunteers.Common.Constants
6. {
7. public class SeedData
8. {
9. public static Organization RunningHearts = new Organization()
10. {
11. Name = "Running Hearts",
12. Address = "Moscow city",
13. Description = "Organization that helps children",
14. Email = "running.hearts@email.com",
15. Phone = "+79990009900"
16. };
17. public static Organization NewYorkHelpers = new Organization()
18. {
19. Name = "NY Helpers",
20. Address = "New York",
21. Description = "Helps everybody in new york",
22. Email = "new.york.helpers@email.com",
23. Phone = "+71112223344"
24. };
25. public static Organization HelpCreators = new Organization()
26. {
27. Name = "HelpCreators",
28. Address = "London",
29. Description = "We are very brave!",
30. Email = "london.help.creators@email.com",
31. Phone = "+73333333333"
32. };
33. }
34. }

## Файл Volunteers.Common.Helpers.IEnumHelper.cs

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. namespace Volunteers.Common.Helpers
6. {
7. public static class IEnumerableHelpers
8. {
9. public static bool IsNullOrEmpty(this IEnumerable<object> list)
10. {
11. if (list == null)
12. {
13. return true;
14. }
15. return !list.Any();
16. }
17. }
18. }

## Файл Volunteers.Common.Models.User.cs

1. using System;
2. using System.Collections.Generic;
3. using System.Text;
4. using Microsoft.AspNetCore.Identity;
5. namespace Volunteers.Common.Models
6. {
7. public class User : IdentityUser
8. {
9. public decimal PointsRate { get; set; }
10. public int Level
11. {
12. get
13. {
14. if (PointsRate == 0)
15. {
16. return 1;
17. }
18. return (int)PointsRate / 20;
19. }
20. }
21. public User()
22. {
23. }
24. public User(string userName)
25. : base(userName)
26. {
27. }
28. }
29. }

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |