Приложения

Приложение 1 – HomeController.cs

```
using BarcodeSenderSystem.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;

namespace BarcodeSenderSystem.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;

        public HomeController(ILogger<HomeController> logger)
        {
            _logger = logger;
        }

        public IActionResult Index()
        {
            return View();
        }

        public IActionResult Privacy()
        {
            return View();
        }

        [ResponseCache(Duration = 0, Location =
ResponseCacheLocation.None, NoStore = true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId =
Activity.Current?.Id ?? HttpContext.TraceIdentifier });
        }
    }
}
```

Приложение 2 – RegisterController.cs

```
using BarcodeSenderSystem.InputModels.Register;
using BarcodeSenderSystem.Services;
using Microsoft.AspNetCore.Mvc;
using System;
```

```
namespace BarcodeSenderSystem.Controllers
{
    public class RegisterController : Controller
    {
        private readonly ISertificateGenerator
sertificateGenerator;
        private readonly IEmailSenderService
emailSenderService;
        private readonly IUserService userService;
        private readonly ICodeService codeService;

        public RegisterController(ISertificateGenerator
sertificateGenerator, IEmailSenderService emailSenderService,
ICodeService qrCodeService, IUserService userService)
        {
            this.sertificateGenerator = sertificateGenerator;
            this.emailSenderService = emailSenderService;
            this.userService = userService;
            this.codeService = qrCodeService;
        }

        [HttpGet]
        public IActionResult Index(string userId = null)
        {

            if (userId != null)
            {
                int id = int.Parse(userId);
                RegisterInputModel model =
userService.GetRegisterData(id);
                return View(model);
            }

            return View();
        }

        [HttpPost]
        public IActionResult Index(RegisterInputModel
inputModel)
        {
            if (!ModelState.IsValid)
            {
                return View(inputModel);
            }
            var baseUrl =
$"{Request.Scheme}://{Request.Host.Value.ToString()}{Request.P
athBase.Value.ToString()}";
            codeService.CreateQrCode(baseUrl);
            codeService.CreateBarCode(baseUrl);
            sertificateGenerator.GenerateSertificate();
```

```
            string fullName = inputModel.FirstName + " " +
inputModel.LastName;
            emailSenderService.SendEmail(inputModel.Email,
fullName);

            return RedirectToAction(nameof(Registered));
        }

        public IActionResult Registered()
        {
            return View();
        }


    }
}
```

Приложение 3 – SummaryController.cs

```
using BarcodeSenderSystem.InputModels.Register;
using BarcodeSenderSystem.Services;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;

namespace BarcodeSenderSystem.Controllers
{
    public class SummaryController : Controller
    {
        private readonly IUserService userService;

        public SummaryController(IUserService userService)
        {
            this.userService = userService;
        }

        [HttpGet]
        public IActionResult Contacts()
        {
            return View();
        }

        [Authorize]
        [HttpGet]
        public IActionResult Initiatives()
        {
            return View();
```

```
        }

        [HttpGet]
        public IActionResult Other()
        {
            return View();
        }

        [HttpGet]
        public IActionResult Apply()
        {
            if (HttpContext.User.Identity.IsAuthenticated)
            {
                int userId =
int.Parse(HttpContext.User.Identity.Name);
                return RedirectToAction("Index", "Register",
new { userId = userId });

            }

            return RedirectToAction("Index", "Register");
        }
    }
}
```

Приложение 4 – UserController.cs

```
using BarcodeSenderSystem.InputModels.User;

using BarcodeSenderSystem.Services;

using Microsoft.AspNetCore.Authentication;
using Microsoft.AspNetCore.Authentication.Cookies;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;

namespace BarcodeSenderSystem.Controllers
{
    public class UserController : Controller
    {
        private readonly IUserService userService;

        public UserController(IUserService userService)
        {
            this.userService = userService;
        }
```

```
        [HttpGet]
        public IActionResult RegisterUser()
        {
            return View();
        }

        [HttpPost]
        public IActionResult
RegisterUser(RegisterUserInputModel inputModel)
        {
            if
(!inputModel.ConfirmPassword.Equals(inputModel.Password))
            {
                ModelState.AddModelError("password", "Password
and confirm password must be equal");
                ModelState.AddModelError("confirmPassword",
"Confirm password and password mu be equal");
            }

            if (!ModelState.IsValid)
            {
                return View(inputModel);
            }

            userService.RegisterUser(inputModel);

            return RedirectToAction("Index","Home");
        }

        [HttpGet]
        public IActionResult Login()
        {
            return View();
        }

        [HttpPost]
        public async Task<IActionResult> Login(LoginInputModel
inputModel)
        {
            if (!userService.IsValidUser(inputModel))
            {
                ModelState.AddModelError("Key", "Invalid
username and password combination!");
            }

            if (!ModelState.IsValid)
            {
                return View(inputModel);
            }
```

```
            int userId =
userService.GetUserId(inputModel.UserName);

            var claims = new List<Claim>
            {
                new Claim(ClaimTypes.Name, userId.ToString()),
                new Claim(ClaimTypes.Role, "User"),
            };

            var claimsIdentity = new ClaimsIdentity(claims,
CookieAuthenticationDefaults.AuthenticationScheme);
            var authProperties = new
AuthenticationProperties();

            await
HttpContext.SignInAsync(CookieAuthenticationDefaults.Authentic
ationScheme,
                new ClaimsPrincipal(claimsIdentity),
                authProperties);

            return RedirectToAction("Index", "Home");
        }

        public async Task<IActionResult> Logout()
        {
            if (HttpContext.User.Identity.IsAuthenticated)
            {
                await
HttpContext.SignOutAsync(CookieAuthenticationDefaults.Authenti
cationScheme);
            }

            return RedirectToAction("Index", "Home");
        }
    }
}
```

Приложение 5 – RegisterInputModel.cs

```
using System.ComponentModel.DataAnnotations;

namespace BarcodeSenderSystem.InputModels.Register
{
    public class RegisterInputModel
    {
        [Required]
        [EmailAddress]
        public string Email { get; set; }

        [Required]
```

```
        public string PhoneNumber { get; set; }

        [Required]
        [StringLength(40, MinimumLength = 3)]
        public string FirstName { get; set; }

        [Required]
        [StringLength(40, MinimumLength = 3)]
        public string LastName { get; set; }
    }
}
```

Приложение 6 – LoginInputModel.cs

```
using System.ComponentModel.DataAnnotations;

namespace BarcodeSenderSystem.InputModels.User
{
    public class LoginInputModel
    {
        [Required]
        [Display(Name = "Потребителско име")]
        public string UserName { get; set; }

        [Required]
        [Display(Name = "Парола")]
        public string Password { get; set; }
    }
}
```

Приложение 7 – RegisterUserInputModel.cs

```
using System.ComponentModel.DataAnnotations;

namespace BarcodeSenderSystem.InputModels.User
{
    public class RegisterUserInputModel
    {
        [Required]
        [Display(Name = "Потребителско име")]
        [StringLength(100, MinimumLength = 3, ErrorMessage =
"{0} трябва да бъде между {2} и {1} символа!")]
        public string UserName { get; set; }

        [Required]
        [Display(Name = "Парола")]
        [StringLength(10, MinimumLength = 4, ErrorMessage =
"{0} трябва да бъде между {2} и {1} символа!")]
        public string Password { get; set; }
```

```
        [Required]
        [Display(Name = "Повтори парола")]
        [StringLength(10, MinimumLength = 4, ErrorMessage =
"{0} трябва да бъде между {2} и {1} символа!")]
        public string ConfirmPassword { get; set; }

        [Required]
        [EmailAddress]
        public string Email { get; set; }

        [Required]
        [Display(Name = "Телефонен номер")]
        [StringLength(14, MinimumLength = 6, ErrorMessage =
"{0} трябва да бъде между {2} и {1} символа!")]
        public string PhoneNumber { get; set; }

        [Required]
        [Display(Name = "Малко име")]
        [StringLength(30, MinimumLength = 3, ErrorMessage =
"{0} трябва да бъде между {2} и {1} символа!")]
        public string FirstName { get; set; }

        [Required]
        [Display(Name = "Фамилно име")]
        [StringLength(30, MinimumLength = 3, ErrorMessage =
"{0} трябва да бъде между {2} и {1} символа!")]
        public string LastName { get; set; }
    }
}
```

Приложение 8 – sertificate.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
</head>
<body style="border-style: groove; border-color: lightgray;
border-radius: 7px; padding: 5px; border-width: 2px;">

    <div style="text-align: center">
        <h2>Сертификат</h2>
        <p>
            Главната роля в кампанията на зелена България „Да
изчистим България заедно“ е поверена на гражданите,
            които с действията и начина си на мислене, се
превръщат от доброволци за един ден в неизменни посланици на
```

```
            зелената идея и каузата да почистим и облагородим
страната си.

            Благодарим ,че се присъединихте.
        </p>


        <img
src="C:\Users\user\Downloads\BarcodeSenderSystem\BarcodeSender
System\bin\Debug\netcoreapp3.1\Barcodes\QrUrl.jpg"
style="width:250px; height:250px;" />
        <img style="margin-bottom: 15px;"
src="C:\Users\user\Downloads\BarcodeSenderSystem\BarcodeSender
System\bin\Debug\netcoreapp3.1\Barcodes\Barcode.jpg"
style="width:300px; height: 150px;" />
    </div>




</body>
</html>
```

Приложение 9 – CodeService.cs

```
using BarcodeLib;
using QRCoder;
using System.Drawing;
using System.Drawing.Imaging;
using System.IO;
using System.Reflection;

namespace BarcodeSenderSystem.Services
{
    public class CodeService : ICodeService
    {
        private readonly string CURRENT_DIRECTORY;
        private const string BARCODE_FOLDER_NAME = "Barcodes";

        public CodeService()
        {
            CURRENT_DIRECTORY =
Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location
) ;
        }

        public void CreateBarCode(string url)
        {
            if (!HasBarcodeFolder())
            {
                CreateBarcodeFolder();
```

```
            }

            Barcode barcodeApi = new Barcode();
            int width = 500;
            int height = 250;
            Color foreColor = Color.Black;
            Color backColor = Color.White;

            Image barcode = barcodeApi.Encode(TYPE.CODE128,
url, foreColor, backColor, width, height);
            const string barcodeImageName = "Barcode.jpg";
            barcode.Save(Path.Combine(CURRENT_DIRECTORY,
BARCODE_FOLDER_NAME, barcodeImageName), ImageFormat.Jpeg);
        }

        public void CreateQrCode(string url)
        {
            if (!HasBarcodeFolder())
            {
                CreateBarcodeFolder();
            }

            QRCodeGenerator qrGenerator = new
QRCodeGenerator();
            QRCodeData data = qrGenerator.CreateQrCode(url,
QRCodeGenerator.ECCLevel.Q);
            QRCode qrCode = new QRCode(data);
            Bitmap arCodeImage = qrCode.GetGraphic(20);
            const string imageName = "QrUrl.jpg";
            arCodeImage.Save(Path.Combine(CURRENT_DIRECTORY,
BARCODE_FOLDER_NAME, imageName),ImageFormat.Jpeg);
        }

        private bool HasBarcodeFolder()
        {
            return
Directory.Exists(Path.Combine(CURRENT_DIRECTORY,
BARCODE_FOLDER_NAME));
        }

        private void CreateBarcodeFolder()
        {

Directory.CreateDirectory(Path.Combine(CURRENT_DIRECTORY,
BARCODE_FOLDER_NAME)); ;
        }
    }
}
```

Приложение 10 – EmailSenderService.cs

```
namespace BarcodeSenderSystem.Services
{
    using System.Net.Mail;
    using System.Net;
    using System.Net.Mime;

    public class EmailSenderService : IEmailSenderService
    {
        private readonly ISertificateGenerator
sertificateGenerator;

        public EmailSenderService(ISertificateGenerator
sertificateGenerator)
        {
            this.sertificateGenerator = sertificateGenerator;
        }

        public void SendEmail(string emailAddress, string
name)
        {
            var fromAddress = new
MailAddress("autoSenderHome@gmail.com", "Sender bot");
            var toAddress = new MailAddress(emailAddress,
name);
            const string fromPassword = "1234a.78";
            const string subject = "Благодаря, че се
присъединихте";
            string body = $"Здравейте {name}. Благодаря Ви ,че
станахте част от нашата инициатива";
            ContentType contentType = new ContentType();
            contentType.MediaType =
MediaTypeNames.Application.Pdf;
            contentType.Name = "Sertificate.pdf";

            var smtp = new SmtpClient
            {
                Host = "smtp.gmail.com",
                Port = 587,
                EnableSsl = true,
                DeliveryMethod = SmtpDeliveryMethod.Network,
                UseDefaultCredentials = false,
                Credentials = new
NetworkCredential(fromAddress.Address, fromPassword)
            };

            using (var message = new MailMessage(fromAddress,
toAddress)
            {
                Subject =subject,
                Body = body
            })
```

```
            {
                string filePath =
sertificateGenerator.GetSertificatePath();

                message.Attachments.Add(new
Attachment(filePath, contentType));
                smtp.Send(message);
            }
        }


    }
}
```

Приложение 11 – ICodeService.cs

```
namespace BarcodeSenderSystem.Services
{
    public interface ICodeService
    {
        void CreateQrCode(string url);

        void CreateBarCode(string url);
    }
}
```

Приложение 12 – IEmailSenderService.cs

```
namespace BarcodeSenderSystem.Services
{
    public interface IEmailSenderService
    {
        void SendEmail(string emailAddress, string name);
    }
}
```

Приложение 13 – ISertificateGenerator.cs

```
namespace BarcodeSenderSystem.Services
{
    public interface ISertificateGenerator
    {
        void GenerateSertificate();

        string GetSertificatePath();
    }
}
```

Приложение 14 – IUserService.cs

```
using BarcodeSenderSystem.InputModels.Register;
using BarcodeSenderSystem.InputModels.User;

namespace BarcodeSenderSystem.Services
{
    public interface IUserService
    {
        void RegisterUser(RegisterUserInputModel inputModel);

        bool IsValidUser(LoginInputModel inputModel);

        int GetUserId(string userName);

        RegisterInputModel GetRegisterData(int id);

    }
}
```

Приложение 15 – SertificateGenerator.cs

```
namespace BarcodeSenderSystem.Services
{
    using System;
    using iText.Html2pdf;
    using System.IO;
    using System.Reflection;

    public class SertificateGenerator : ISertificateGenerator
    {
        private readonly string CURRENT_DIRECTORY;
        private const string FOLDER_NAME = "Sertificate";
        private const string FILE_NAME = "Sertificate.pdf";

        public SertificateGenerator()
        {
            CURRENT_DIRECTORY =
Path.GetDirectoryName(Assembly.GetExecutingAssembly().Location
);
        }

        public void GenerateSertificate()
        {
            if (!HasDirectory())
            {
                CreateDirectory();
            }
```

```
            HtmlConverter.ConvertToPdf(
                new
FileInfo(@"SertificateHtml\sertificate.html"),
                new FileInfo(Path.Combine(CURRENT_DIRECTORY,
FOLDER_NAME, FILE_NAME)));
        }

        private bool HasDirectory()
        {
            return
Directory.Exists(Path.Combine(CURRENT_DIRECTORY,
FOLDER_NAME));
        }

        private void CreateDirectory()
        {

Directory.CreateDirectory(Path.Combine(CURRENT_DIRECTORY,
FOLDER_NAME));
        }

        public string GetSertificatePath()
        {
            return Path.Combine(CURRENT_DIRECTORY,
FOLDER_NAME, FILE_NAME);
        }
    }
}
```

Приложение 16 – UserService.cs

```
using BarcodeSenderSystem.InputModels.Register;
using BarcodeSenderSystem.InputModels.User;
using System;
using System.IO;
using System.Linq;

namespace BarcodeSenderSystem.Services
{
    public class UserService : IUserService
    {
        private const string USERS_FOLDER_NAME = "users";
        private const string USERS_FILE_NAME = "users.txt";
        private const string SEPPARATOR = ", ";

        private readonly string fullPath;

        public UserService()
        {
```

```
            fullPath = Path.Combine(USERS_FOLDER_NAME,
USERS_FILE_NAME);
            if (!HasUserFile())
            {
                CreateUserFile();
            }
        }

        public bool IsValidUser(LoginInputModel inputModel)
        {
            string userName = inputModel.UserName;
            string password = inputModel.Password;


            foreach (string line in
File.ReadAllLines(fullPath))
            {
                string[] data =
line.Split(SEPPARATOR).ToArray();
                string currentUserName = data[1];
                string currentPassword = data[2];

                bool isValidUser =
currentUserName.Equals(userName);
                bool isValidPassword =
currentPassword.Equals(password);

                if (isValidUser && isValidPassword)
                {
                    return true;
                }

            }

            return false;
        }

        public void RegisterUser(RegisterUserInputModel
inputModel)
        {
            //id, UserName, Password, Email, PhoneNumber,
FirstName, LastName
            //Validate username no duplicates
            //Validate email no duplicates
            //Validate phoneNumber no duplicates
            int id = GenUserId();
            string record = $"{id}, {inputModel.UserName},
{inputModel.Password}, {inputModel.Email},
{inputModel.PhoneNumber}, {inputModel.FirstName},
{inputModel.LastName}";
```

```
            File.AppendAllText(fullPath, record +
Environment.NewLine);
        }

        public int GetUserId(string userName)
        {
            foreach (string line in
File.ReadAllLines(fullPath))
            {
                string[] data =
line.Split(SEPPARATOR).ToArray();
                string currentUserName = data[1];

                if (currentUserName == userName)
                {
                    int userId = int.Parse(data[0]);
                    return userId;
                }
            }

            return -1;
        }

        public RegisterInputModel GetRegisterData(int id)
        {
            RegisterInputModel inputModel = null;

            foreach (string line in
File.ReadAllLines(fullPath))
            {
                string[] data =
line.Split(SEPPARATOR).ToArray();
                int currentId = int.Parse(data[0]);

                if (currentId == id)
                {
                    string email = data[3];
                    string phoneNumber = data[4];
                    string firstName = data[5];
                    string lastName = data[6];
                    inputModel = new RegisterInputModel
                    {
                        Email = email,
                        FirstName = firstName,
                        LastName = lastName,
                        PhoneNumber = phoneNumber,
                    };

                    return inputModel;
                }
            }
```

```
            return inputModel;
        }

        private int GenUserId()
        {
            string[] data = null;
            bool hasLines = false;
            foreach (string line in
File.ReadAllLines(fullPath))
            {
                hasLines = true;
                data = line.Split(", ").ToArray();
            }

            int lastUserId = 0;
            if (hasLines)
            {
                lastUserId = int.Parse(data[0]);
            }

            return ++lastUserId;

        }

        private void CreateUserFile()
        {
            if (!HasUserFolder())
            {
                Directory.CreateDirectory(USERS_FOLDER_NAME);
            }

            if (!HasUserFile())
            {
                File.Create(fullPath);
            }
        }

        private bool HasUserFile()
        {
            return File.Exists(fullPath);
        }

        private bool HasUserFolder()
        {
            return Directory.Exists(USERS_FOLDER_NAME);
        }

    }
}
```

Приложение 17 – Index.cshtml

```
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <h1 class="display-4">Добре Дошли</h1>
    <br />
    <p style="font-size: 1.3rem;">Вярваме, че даването на
личен пример и поддържането на
    чиста природа са възможни всеки ден - без заплаха за
здравето на хората.
    Затова през 2022 г.,
    стремежът на инициативата е да съхрани постигнатото през
годините и да го надгради,
    като мотивира хората за още повече нови добри дела.</p>

    <p style="font-size: 1.3rem;">Стани част от нашата
инициатива като се регистринраш на <a asp-
controller="Register" asp-action="Index">този</a> адрес</p>

    <div class="container">
        <div class="row">
            <div class="col">
                <img style="width: 100%; height: 85%;"
src="https://cdn2.vectorstock.com/i/1000x1000/23/36/people-
cleaning-park-community-team-collecting-vector-34332336.jpg"
alt="Alternate Text" />
            </div>
            <div class="col">
                <img style="width: 100%; height: auto;"
src="https://previews.123rf.com/images/dolgachov/dolgachov1609
/dolgachov160903885/62354044-volunteering-charity-cleaning-
people-and-ecology-concept-group-of-happy-volunteers-with-
garbage-bags.jpg" alt="Alternate Text" />
            </div>
            <div class="col">
                <img style="width: 100%; height: auto;"
src="https://media.istockphoto.com/photos/people-cleaning-the-
environment-picture-id1077156290" alt="Alternate Text" />
            </div>
        </div>
    </div>
</div>
```

Приложение 18 – Privacy.cshtml

```
@{
    ViewData["Title"] = "За нас";
```

```
}
<h1>@ViewData["Title"]</h1>

<p>
    Нашите инициативи са достъпни за всички, които искат да се
присъединят. <br/>
    Организацията ни е създадена още през далечната 2022
година.
</p>
```

Приложение 17 – Index.cshtml

```
@model BarcodeSenderSystem.InputModels.Register.RegisterInputModel

<h1>Register</h1>
<form asp-controller="Register" asp-action="Index" method="post"
style="border-style: groove;border-width: 1px; border-radius:7px;
padding: 1%; border-color: lightgray;">
    <div class="form-group">
        <div class="row">
            <div class="col">
                <label>Имейл</label>
                <input asp-for="@Model.Email" class="form-control"
/>
                <div class="text-danger">
                    <span asp-validation-for="@Model.Email"></span>
                </div>
            </div>
            <div class="col">
                <label>Телефонен номер</label>
                <input asp-for="@Model.PhoneNumber" class="form-
control" />
                <div class="text-danger">
                    <span asp-validation-
for="@Model.PhoneNumber"></span>
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col">
                <label>Първо име</label>
                <input asp-for="@Model.FirstName" class="form-
control" />
                <div class="text-danger">
                    <span asp-validation-
for="@Model.FirstName"></span>
                </div>
```

```
            </div>
            <div class="col">
                <label>Фамилно име</label>
                <input asp-for="@Model.LastName" class="form-
control" />
                <div class="text-danger">
                    <span asp-validation-
for="@Model.LastName"></span>
                </div>
            </div>
        </div>
    </div>
    <button type="submit" class="btn btn-
success">Кандидатствай</button>

</form>
```

Приложение 18 – Registered.cshtml

```
<h1 class="text-success">Благодарим Ви ,че станахте част от
нашата инициатива.</h1>
<a asp-controller="Home" asp-action="Index">Главна
страница</a>
```

Приложение 19 – _Layout.cshtml

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>@ViewData["Title"] - Зелена България</title>
    <link rel="stylesheet"
href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" />
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-
sm navbar-light bg-white border-bottom box-shadow mb-3">
            <div class="container">
                <a class="navbar-brand" asp-area="" asp-
controller="Home" asp-action="Index">Зелена България</a>
                <button class="navbar-toggler" type="button"
data-toggle="collapse" data-target=".navbar-collapse" aria-
controls="navbarSupportedContent"
                        aria-expanded="false" aria-
label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
```

```
                <div class="navbar-collapse collapse d-sm-
inline-flex flex-sm-row-reverse">
                    @if (User.Identity.IsAuthenticated)
                    {
                        <ul class="navbar-nav flex-grow-1
float-right" style="margin-left: 30%;">
                            <li class="nav-item">
                                <a class="nav-link text-dark"
asp-area="" asp-controller="User" asp-action="Logout">Отпиши
се</a>
                            </li>
                        </ul>
                    }
                    else
                    {
                        <ul class="navbar-nav flex-grow-1
float-right" style="margin-left: 20%;">
                            <li class="nav-item">
                                <a class="nav-link text-dark"
asp-area="" asp-controller="User" asp-
action="RegisterUser">Регистрирай се</a>
                            </li>
                            <li class="nav-item">
                                <a class="nav-link text-dark"
asp-area="" asp-controller="User" asp-
action="Login">Вписване</a>
                            </li>
                        </ul>
                    }

                    <ul class="navbar-nav flex-grow-1">
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-
area="" asp-controller="Home" asp-action="Index">Начало</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-
area="" asp-controller="Register" asp-
action="Index">Кандидатствай </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-
area="" asp-controller="Summary" asp-
action="Initiatives">Инициативи </a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-
area="" asp-controller="Summary" asp-
action="Contacts">Контакти</a>
                        </li>
                        <li class="nav-item">
```

```
                            <a class="nav-link text-dark" asp-
area="" asp-controller="Home" asp-action="Privacy">За нас</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-
area="" asp-controller="Summary" asp-action="Other">Други</a>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>
    </header>
    <div class="container">
        <main role="main" class="pb-3">
            @RenderBody()
        </main>
    </div>

    <footer class="border-top footer text-muted">
        <div class="container">
            &copy; 2021 - BarcodeSenderSystem - <a asp-area=""
asp-controller="Home" asp-action="Privacy">Privacy</a>
        </div>
    </footer>
    <script src="~/lib/jquery/dist/jquery.min.js"></script>
    <script
src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script
>
    <script src="~/js/site.js" asp-append-
version="true"></script>
    @RenderSection("Scripts", required: false)
</body>
</html>
```

Приложение 20 – _ValidationScriptsPartial.cshtml

```
<script src="~/lib/jquery-
validation/dist/jquery.validate.min.js"></script>
<script src="~/lib/jquery-validation-
unobtrusive/jquery.validate.unobtrusive.min.js"></script>
```

Приложение 21 – Error.cshtml

```
@model ErrorViewModel
@{
    ViewData["Title"] = "Error";
}

<h1 class="text-danger">Error.</h1>
```

```
<h2 class="text-danger">An error occurred while processing
your request.</h2>

@if (Model.ShowRequestId)
{
    <p>
        <strong>Request ID:</strong>
<code>@Model.RequestId</code>
    </p>
}

<h3>Development Mode</h3>
<p>
    Swapping to <strong>Development</strong> environment will
display more detailed information about the error that
occurred.
</p>
<p>
    <strong>The Development environment shouldn't be enabled
for deployed applications.</strong>
    It can result in displaying sensitive information from
exceptions to end users.
    For local debugging, enable the
<strong>Development</strong> environment by setting the
<strong>ASPNETCORE_ENVIRONMENT</strong> environment variable
to <strong>Development</strong>
    and restarting the app.
</p>
```

Приложение 22 – Contacts.cshtml

```
<h1>Контакти</h1>

<ul>
    <li>Email: example@email.com</li>
    <li>Phone number: example phone number</li>
    <li>City: example city</li>
    <li>Address: example address</li>
</ul>
```

Приложение 23 – Initiatives.cshtml

```
<h1 class="text-center">Инициативи</h1>

<table class="table table-striped">
    <thead>
        <tr>
            <th style="width:70%;" scope="row">Инициатива</th>
            <th style="width: 30%;" scope="row">Действие</th>
        </tr>
    </thead>
```

```
    <tbody>
        <tr>
            <th style="width:70%;">Благотворителен
концерт</th>
            <th style="width: 30%;"><a class="btn btn-outline-
success" asp-controller="Summary" asp-action="Apply"
rel="stylesheet">Кандидатствай</a></th>
        </tr>
        <tr>
            <th style="width:70%;">Изкачване на връх
Мусала</th>
            <th style="width: 30%;"><a class="btn btn-outline-
success" asp-controller="Summary" asp-action="Apply"
rel="stylesheet">Кандидатствай</a></th>
        </tr>
        <tr>
            <th style="width:70%;">Изкачване на връх
Ботев</th>
            <th style="width: 30%;"><a class="btn btn-outline-
success" asp-controller="Summary" asp-action="Apply"
rel="stylesheet">Кандидатствай</a></th>
        </tr>
        <tr>
            <th style="width:70%;"> Шоу с коли</th>
            <th style="width: 30%;"><a class="btn btn-outline-
success" asp-controller="Summary" asp-action="Apply"
rel="stylesheet">Кандидатствай</a></th>
        </tr>
        <tr>
            <th style="width:70%;">Благотворителен тятър</th>
            <th style="width: 30%;"><a class="btn btn-outline-
success" asp-controller="Summary" asp-action="Apply"
rel="stylesheet">Кандидатствай</a></th>
        </tr>

    </tbody>
</table>
```

Приложение 24 – Other.cshtml

```
<h1>Други</h1>
<hr />
<h4>Партньори</h4>

<ul>
    <li>Партньор 1 - <a href="#">example link</a></li>
    <li>Партньор 2 - <a href="#">example link</a></li>
    <li>Партньор 3 - <a href="#">example link</a></li>
    <li>Партньор 4 - <a href="#">example link</a></li>
</ul>
```

Приложение 25 – Login.cshtml

```
@model BarcodeSenderSystem.InputModels.User.LoginInputModel

<form asp-controller="User" asp-action="Login" method="post">
    <div class="text-danger" asp-validation-
summary="All"></div>

    <div class="form-group">
        <label asp-for="@Model.UserName"></label>
        <input class="form-control" asp-for="@Model.UserName"
/>
        <div class="text-danger">
            <span asp-validation-for="@Model.UserName"></span>
        </div>
    </div>
    <div class="form-group">
        <label asp-for="@Model.Password"></label>
        <input type="password" class="form-control" asp-
for="@Model.Password" />
        <div class="text-danger">
            <span asp-validation-for="@Model.Password"></span>
        </div>
    </div>
    <p>Нямате регистрация? <a asp-controller="User" asp-
action="RegisterUser">Регистрирайте се на този адрес</a></p>
    <button class="btn btn-success" type="submit">Впиши
се</button>
</form>
```

Приложение 26 – RegisterUser.cshtml

```
@model
BarcodeSenderSystem.InputModels.User.RegisterUserInputModel
<form asp-controller="User" asp-action="RegisterUser"
method="post">
    <div class="form-group">
        <div class="row">

            <div class="col">
                <label asp-for="@Model.UserName"></label>
                <input class="form-control" asp-
for="@Model.UserName" />
                <div class="text-danger">
                    <span asp-validation-
for="@Model.UserName"></span>
                </div>
            </div>

            <div class="col">

                <label asp-for="@Model.Email"></label>
                <input class="form-control" asp-
for="@Model.Email" />
```

```
                <div class="text-danger">
                    <span asp-validation-
for="@Model.Email"></span>
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col">
                <label asp-for="@Model.Password"></label>
                <input type="password" class="form-control"
asp-for="@Model.Password" />
                <div class="text-danger">
                    <span asp-validation-
for="@Model.Password"></span>
                </div>
            </div>
            <div class="col">
                <label asp-
for="@Model.ConfirmPassword"></label>
                <input type="password" class="form-control"
asp-for="@Model.ConfirmPassword" />
                <div class="text-danger">
                    <span asp-validation-
for="@Model.ConfirmPassword"></span>
                </div>
            </div>
        </div>

        <div class="row">
            <div class="col">
                <label asp-for="@Model.FirstName"></label>
                <input class="form-control" asp-
for="@Model.FirstName" />
                <div class="text-danger">
                    <span asp-validation-
for="@Model.FirstName"></span>
                </div>
            </div>
            <div class="col">
                <label asp-for="@Model.LastName"></label>
                <input class="form-control" asp-
for="@Model.LastName" />
                <div class="text-danger">
                    <span asp-validation-
for="@Model.LastName"></span>
                </div>
            </div>
        </div>

        <div class="row">
```

```
            <div class="col-lg-2">
                <label asp-for="@Model.PhoneNumber"></label>
                <input class="form-control" asp-
for="@Model.PhoneNumber" />
                <div class="text-danger">
                    <span asp-validation-
for="@Model.PhoneNumber"></span>
                </div>
            </div>
        </div>
    </div>

    <button class="btn btn-success" type="submit">Регистрирай
ce</button>
</form>
```

Приложение 27 – _ViewImports.cshtml

```
@using BarcodeSenderSystem
@using BarcodeSenderSystem.Models
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

Приложение 28 – _ViewStart.cshtml

```
@{
    Layout = "_Layout";
}
```

Приложение 29 – appsettings.json

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*"
}
```

Приложение 30 – Program.cs

```
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.Hosting;
using Microsoft.Extensions.Logging;
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Threading.Tasks;

namespace BarcodeSenderSystem
{
    public class Program
    {
        public static void Main(string[] args)
        {
            CreateHostBuilder(args).Build().Run();
        }

        public static IHostBuilder CreateHostBuilder(string[]
args) =>
            Host.CreateDefaultBuilder(args)
                .ConfigureWebHostDefaults(webBuilder =>
                {
                    webBuilder.UseStartup<Startup>();
                });
    }
}
```

Приложение 31 – Startup.cs

```
using BarcodeSenderSystem.Services;
using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Microsoft.AspNetCore.Authentication.Cookies;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace BarcodeSenderSystem
{
    public class Startup
    {
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        public IConfiguration Configuration { get; }

        // This method gets called by the runtime. Use this
method to add services to the container.
```

```
        public void ConfigureServices(IServiceCollection
services)
        {

services.AddAuthentication(CookieAuthenticationDefaults.Authen
ticationScheme)
                .AddCookie(option =>
                {
                    option.LoginPath = "/User/Login";
                });

            services.AddControllersWithViews();
            services.AddTransient<ICodeService,
CodeService>();
            services.AddTransient<IEmailSenderService,
EmailSenderService>();
            services.AddTransient<ISertificateGenerator,
SertificateGenerator>();
            services.AddTransient<IUserService,
UserService>();
        }

        // This method gets called by the runtime. Use this
method to configure the HTTP request pipeline.
        public void Configure(IApplicationBuilder app,
IWebHostEnvironment env)
        {


            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
            }
            else
            {
                app.UseExceptionHandler("/Home/Error");
            }
            app.UseStaticFiles();

            app.UseRouting();

            app.UseAuthentication();
            app.UseAuthorization();
            CookiePolicyOptions options = new
CookiePolicyOptions()
            {
                MinimumSameSitePolicy =
Microsoft.AspNetCore.Http.SameSiteMode.Strict,

            };
            app.UseCookiePolicy(options);
```

```
            app.UseEndpoints(endpoints =>
            {
                endpoints.MapControllerRoute(
                    name: "default",
                    pattern:
"{controller=Home}/{action=Index}/{id?}");
            });
        }
    }
}
```