**15 Платформа разработки Веб-приложений ASP.NET**

Задание 1. Телефонный справочник

Листинг задачи:

using Microsoft.AspNetCore.Mvc;

using PhoneBook.Models;

using PhoneBook.Services;

namespace PhoneBook.Controllers

{

public class ContactsController : Controller

{

private readonly IContactService \_contactService;

public ContactsController(IContactService contactService)

{

\_contactService = contactService;

}

[HttpGet]

public IActionResult Index(string searchTerm)

{

IEnumerable<Contact> contacts;

if (!string.IsNullOrWhiteSpace(searchTerm))

{

contacts = \_contactService.SearchContacts(searchTerm);

ViewData["SearchTerm"] = searchTerm;

}

else

{

contacts = \_contactService.GetAllContacts();

}

if (ViewData["ContactFormData"] == null)

{

ViewData["ContactFormData"] = new ContactViewModel();

}

if (!ViewData.ContainsKey("SearchTerm"))

{

ViewData["SearchTerm"] = null;

}

return View(contacts);

}

[HttpGet("Contacts/Search/{name}")]

public IActionResult Search(string name)

{

return RedirectToAction(nameof(Index), new { searchTerm = name });

}

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult Create([Bind(Prefix = "contactFormData")] ContactViewModel contactViewModel)

{

if (ModelState.IsValid)

{

\_contactService.AddContact(contactViewModel);

TempData["SuccessMessage"] = "Контакт успешно добавлен!";

return RedirectToAction(nameof(Index));

}

var allContacts = \_contactService.GetAllContacts();

ViewData["ContactFormData"] = contactViewModel;

ViewData["SearchTerm"] = null;

return View("Index", allContacts);

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Данные для поиска контактов, Имя контакта, Номер телефона | Таблица контактов, Форма поиска, Форма добавления, Сообщение об ошибке и внесении контакта |

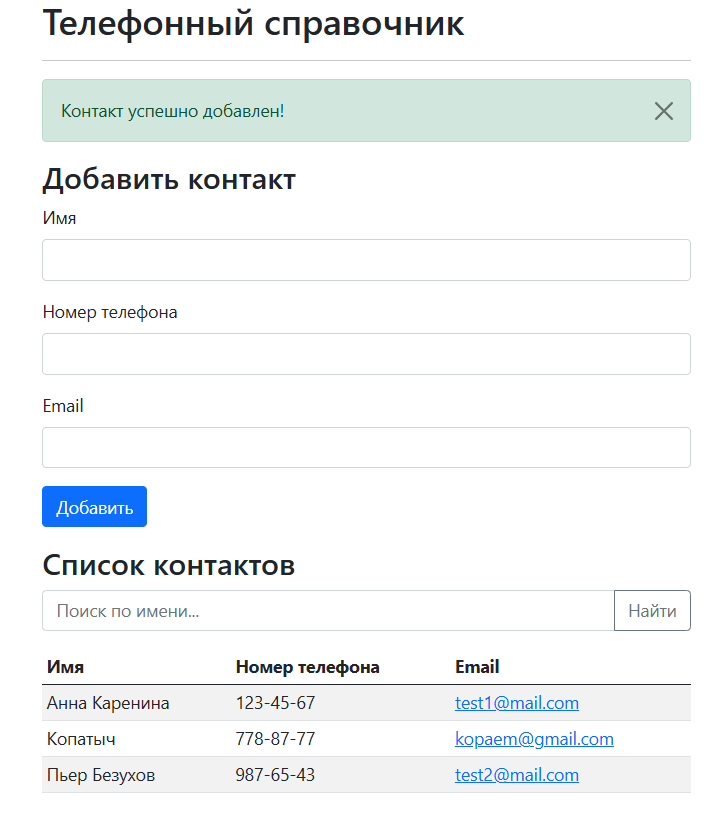
Анализ результатов:

Рисунок 1.1 – Результат работы программы