ПРИЛОЖЕНИЕ Б. ТЕКСТ ПРОГРАММЫ

АНОТАЦИЯ

В данном программном документе приведено текст программы для информационной системы управления в стоматологии «Healthy smile»

В данном программном документе, в разделе «Текст программы» указана информация о приложении, области применения программы, модулей программы и код каждого программного модуля.

ОГЛАВЛЕНИЕ

[1. ТЕКСТ ПРОГРАММЫ 3](#_Toc125899161)

# ТЕКСТ ПРОГРАММЫ

* 1. Наименование программы

Наименование - информационная система управления пациентами в стоматологии «Healthy smile».

* 1. Область применения программы

Приложение предназначено к применению пользователями в клинике стоматологии.

* 1. Модули

Таблица – Модули приложения персонала

| № | Модуль | Описание | Размер (в Кбайтах) | Количество строк кода |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Appointment.cs | Данный класс отвечает за авторизацию пользователя в системе. | 611 | 22 |
| 2 | Clinic.cs | Данный класс отвечает за функционал пользователя с роль «Менеджер по заявкам» | 265 | 15 |
| 3 | DateAppointment.cs | Данный класс отвечает за функционал пользователя с роль «Технический специалист» | 351 | 17 |
| 4 | Employee.cs | Данный класс отвечает за модель используемую при создания запроса. | 645 | 21 |
| 5 | Patient.cs | Данный класс отвечает за модель используемую при создания запроса. | 654 | 22 |
| 6 | Role.cs | Данный класс отвечает за модель используемую при создания запроса. | 255 | 15 |
| 7 | Service.cs | Данный класс отвечает за модель используемую при создания запроса. | 312 | 16 |
| 8 | HomeController.cs | Данный класс отвечает за контроллер для взаимодействия с моделями и апи | 20 | 501 |
| 9 | AddDateApppointment.cs | Данный класс отвечает за представление используемую при создания запроса. | 2,47 | 91 |
| 10 | AddService.cs | Данный класс отвечает за представление используемую при создания запроса. | 2,29 | 85 |
| 11 | AdminDateAppointment.cs | Данный класс отвечает за представление используемую при создания запроса. | 4,08 | 108 |
| 12 | AdminService.cs | Данный класс отвечает за представление используемую при создания запроса. | 2,75 | 82 |
| 13 | Auth.cs | Данный класс отвечает за представление используемую при создания запроса. | 1,90 | 54 |
| 14 | CreateApp.cs | Данный класс отвечает за представление используемую при создания запроса. | 3,74 | 113 |
| 15 | MainDoctor.cs | Данный класс отвечает за представление используемую при создания запроса. | 3,91 | 58 |
| 16 | MainPatient.cs | Данный класс отвечает за представление используемую при создания запроса. | 1,91 | 18 |
| 17 | ProfilePatient.cs | Данный класс отвечает за представление используемую при создания запроса. | 3,98 | 103 |
| 18 | Reg.cs | Данный класс отвечает за представление используемую при создания запроса. | 3,34 | 93 |
| 19 | Service.cs | Данный класс отвечает за представление используемую при создания запроса. | 1,60 | 52 |
| 20 | UpdateAppointment.cs | Данный класс отвечает за представление используемую при создания запроса. | 2,94 | 91 |
| 21 | UpdateDateAppointment | Данный класс отвечает за представление используемую при создания запроса. | 2,42 | 911 |
| 22 | UpdateService | Данный класс отвечает за представление используемую при создания запроса. | 2,48 | 90 |

* 1. Код программы

1. Appointment

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations.Schema;

namespace DentistryWeb.Models

{

public partial class Appointment

{

public int? IdAppointment { get; set; }

public string Number { get; set; } = null!;

public string Complaint { get; set; } = null!;

public string Treatment { get; set; } = null!;

public int EmployeeId { get; set; }

public int PatientId { get; set; }

public int ServiceId { get; set; }

public int DateAppointmentId { get; set; }

}

}

1. Clinic

using System;

using System.Collections.Generic;

namespace DentistryWeb.Models

{

public partial class Clinic

{

public int IdClinic { get; set; }

public string AddresClinic { get; set; } = null!;

}

}

1. DateAppointment

using System;

using System.Collections.Generic;

namespace DentistryWeb.Models

{

public partial class DateAppointment

{

public int IdDateAppointment { get; set; }

public DateTime Date { get; set; }

public TimeSpan Time { get; set; }

public bool Free { get; set; }

}

}

1. Employee

using System;

using System.Collections.Generic;

namespace DentistryWeb.Models

{

public partial class Employee

{

public int IdEmployee { get; set; }

public string SurnameEmployee { get; set; } = null!;

public string FirstNameEmployee { get; set; } = null!;

public string MiddleEmployee { get; set; } = null!;

public string EmailEmployee { get; set; } = null!;

public string PasswordEmployee { get; set; } = null!;

public int Experience { get; set; }

public int RoleId { get; set; }

public int ClinicId { get; set; }

}

}

1. Patient

using System;

using System.Collections.Generic;

namespace DentistryWeb.Models

{

public partial class Patient

{

public int IdPatient { get; set; }

public string SurnamePatient { get; set; } = null!;

public string FirstNamePatient { get; set; } = null!;

public string MiddlePatient { get; set; } = null!;

public string EmailPatient { get; set; } = null!;

public string PasswordPatient { get; set; } = null!;

public string Sex { get; set; } = null!;

public int RoleId { get; set; }

public string Age { get; set; } = null!;

}

}

1. Role

using System;

using System.Collections.Generic;

namespace DentistryWeb.Models

{

public partial class Role

{

public int IdRole { get; set; }

public string NameRole { get; set; } = null!;

}

}

1. Service

using System;

using System.Collections.Generic;

namespace DentistryWeb.Models

{

public partial class Service

{

public int IdService { get; set; }

public string NameService { get; set; } = null!;

public int PriceService { get; set; }

}

}

1. HomeController

using DentistryWeb.Models;

using Microsoft.AspNetCore.Authentication.Cookies;

using Microsoft.AspNetCore.Authentication;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using Newtonsoft.Json;

using System.Diagnostics;

using System.Net.Http;

using System.Security.Claims;

using System.Text;

using DentistryWeb.Models;

using System.Reflection;

using Microsoft.AspNetCore.Authorization;

using System.Net.Http.Headers;

using Dentistry\_API.Models;

namespace DentistryWeb.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

//Авторизация

public IActionResult Auth()

{

return View();

}

[HttpPost]

public async Task<IActionResult> Auth(Patient model)

{

string email = model.EmailPatient;

string password = model.PasswordPatient;

using (var httpClient = new HttpClient())

{

var requestData = new { Email = email, Password = password };

var json = JsonConvert.SerializeObject(requestData);

System.Diagnostics.Debug.WriteLine($"Request JSON: {json}");

var content = new StringContent(json, Encoding.UTF8, "application/json");

var apiResponse = await httpClient.PostAsync("https://localhost:7015/login", content);

if (apiResponse.IsSuccessStatusCode)

{

var responseJson = await apiResponse.Content.ReadAsStringAsync();

System.Diagnostics.Debug.WriteLine($"Response JSON: {responseJson}");

var patient = JsonConvert.DeserializeObject<Patient>(responseJson);

if (patient != null && patient.RoleId == 1)

{

return RedirectToAction("MainPatient", "Home");

}

else if (patient != null && patient.RoleId == 2)

{

return RedirectToAction("MainDoctor", "Home");

}

else if (patient != null && patient.RoleId == 3)

{

return RedirectToAction("AdminService", "Home");

}

}

var responseContent = await apiResponse.Content.ReadAsStringAsync();

System.Diagnostics.Debug.WriteLine($"Response JSON: {responseContent}");

ViewBag.ErrorMessage = "Неверный логин или пароль";

return View("Auth");

}

}

//Регистрация

public ViewResult Reg() => View();

[HttpPost]

public async Task<IActionResult> Reg(Patient patient)

{

Patient recivedpat = new Patient();

using (var httpClient = new HttpClient())

{

StringContent content = new StringContent(JsonConvert.SerializeObject(patient), Encoding.UTF8, "application/json");

using (var response = await httpClient.PostAsync("https://localhost:7015/api/Patients", content))

{

string apiResponse = await response.Content.ReadAsStringAsync();

ViewBag.Result = "Success";

recivedpat = JsonConvert.DeserializeObject<Patient>(apiResponse);

}

}

ViewBag.ErrorMessage = "Введены некорректные данные";

return View("MainPatient");

}

//Окно врача

public async Task<IActionResult> MainDoctor()

{

List<Appointment> appList = new List<Appointment>();

List<Patient> patientList = new List<Patient>();

List<DateAppointment> dateList = new List<DateAppointment>();

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Appointments"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

appList = JsonConvert.DeserializeObject<List<Appointment>>(apiResponse);

}

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Patients"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

patientList = JsonConvert.DeserializeObject<List<Patient>>(apiResponse);

}

using (var response = await httpClient.GetAsync("https://localhost:7015/api/DateAppointments"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

dateList = JsonConvert.DeserializeObject<List<DateAppointment>>(apiResponse);

}

}

var models = new Tuple<List<Appointment>, List<Patient>, List<DateAppointment>>(appList, patientList, dateList);

return View(models);

}

public async Task<IActionResult> UpdateAppointment(int id)

{

Appointment appointment = new Appointment();

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Appointments/" + id))

{

string apiResponse = await response.Content.ReadAsStringAsync();

appointment = JsonConvert.DeserializeObject<Appointment>(apiResponse);

}

}

return View(appointment);

}

[HttpPost]

public async Task<IActionResult> UpdateAppointment(Appointment appointment, int id)

{

Appointment receivedapp = new Appointment();

using (var httpClient = new HttpClient())

{

StringContent content = new StringContent(JsonConvert.SerializeObject(appointment), Encoding.UTF8, "application/json");

using (var response = await httpClient.PutAsync("https://localhost:7015/api/Appointments/" + id, content))

{

string apiResponse = await response.Content.ReadAsStringAsync();

ViewBag.Result = "Success";

receivedapp = JsonConvert.DeserializeObject<Appointment>(apiResponse);

}

}

return View(receivedapp);

}

[HttpPost]

public async Task<IActionResult> DeleteAppointment(int IdAppointment)

{

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.DeleteAsync("https://localhost:7015/api/Appointments/" + IdAppointment))

{

string apiResponse = await response.Content.ReadAsStringAsync();

}

}

return RedirectToAction("MainDoctor");

}

//Окно Пациента

public IActionResult MainPatient()

{

return View();

}

//Вывод в списки вариантов записи

public async Task<IActionResult> CreateApp()

{

List<Service> serviceList = new List<Service>();

List<Employee> empList = new List<Employee>();

List<DateAppointment> dateList = new List<DateAppointment>();

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Services"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

serviceList = JsonConvert.DeserializeObject<List<Service>>(apiResponse);

}

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Employees"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

empList = JsonConvert.DeserializeObject<List<Employee>>(apiResponse);

}

using (var response = await httpClient.GetAsync("https://localhost:7015/api/DateAppointments"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

dateList = JsonConvert.DeserializeObject<List<DateAppointment>>(apiResponse);

}

}

var models = new Tuple<List<Service>, List<Employee>, List<DateAppointment>>(serviceList, empList, dateList);

return View(models);

}

//Создание записи

[HttpPost, ActionName("CreateApp")]

public async Task<IActionResult> HandleCreateApp(Appointment app)

{

Appointment recivedapp = new Appointment();

using (var httpClient = new HttpClient())

{

StringContent content = new StringContent(JsonConvert.SerializeObject(app), Encoding.UTF8, "application/json");

using (var response = await httpClient.PostAsync("https://localhost:7015/api/Appointments", content))

{

string apiResponse = await response.Content.ReadAsStringAsync();

recivedapp = JsonConvert.DeserializeObject<Appointment>(apiResponse);

System.Diagnostics.Debug.WriteLine($"Response JSON: {apiResponse}");

}

}

TempData["SuccessMessage"] = "Запись успешно создана";

return RedirectToAction("CreateApp", "Home");

}

//Профиль Пациента

public async Task<IActionResult> ProfilePatient()

{

List<Appointment> appList = new List<Appointment>();

List<Service> servicaList = new List<Service>();

List<Employee> empList = new List<Employee>();

List<DateAppointment> dateList = new List<DateAppointment>();

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Appointments"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

appList = JsonConvert.DeserializeObject<List<Appointment>>(apiResponse);

}

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Services"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

servicaList = JsonConvert.DeserializeObject<List<Service>>(apiResponse);

}

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Employees"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

empList = JsonConvert.DeserializeObject<List<Employee>>(apiResponse);

}

using (var response = await httpClient.GetAsync("https://localhost:7015/api/DateAppointments"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

dateList = JsonConvert.DeserializeObject<List<DateAppointment>>(apiResponse);

}

}

var models = new Tuple<List<Appointment>, List<Service>, List<Employee>, List<DateAppointment>> (appList,servicaList, empList, dateList);

return View(models);

}

//Отмена записи

[HttpPost]

public async Task<IActionResult> DeleteAppointmentPatient(int IdAppointment)

{

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.DeleteAsync("https://localhost:7015/api/Appointments/" + IdAppointment))

{

string apiResponse = await response.Content.ReadAsStringAsync();

}

}

return RedirectToAction("ProfilePatient");

}

//Вывод услуг Пациент

public async Task<IActionResult> Service()

{

List<Service> serviceList = new List<Service>();

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Services"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

serviceList = JsonConvert.DeserializeObject<List<Service>>(apiResponse);

}

}

return View(serviceList);

}

//Окно Админа

//Вывод услуг Админ

public async Task<IActionResult> AdminService()

{

List<Service> serviceList = new List<Service>();

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Services"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

serviceList = JsonConvert.DeserializeObject<List<Service>>(apiResponse);

}

}

return View(serviceList);

}

public ViewResult AddService() => View();

[HttpPost]

public async Task<IActionResult> AddService(Service service)

{

Service recivedservice = new Service();

using (var httpClient = new HttpClient())

{

StringContent content = new StringContent(JsonConvert.SerializeObject(service), Encoding.UTF8, "application/json");

using (var response = await httpClient.PostAsync("https://localhost:7015/api/Services", content))

{

string apiResponse = await response.Content.ReadAsStringAsync();

ViewBag.Result = "Success";

recivedservice = JsonConvert.DeserializeObject<Service>(apiResponse);

}

}

return View(recivedservice);

}

public async Task<IActionResult> UpdateService(int id)

{

Service service = new Service();

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.GetAsync("https://localhost:7015/api/Services/" + id))

{

string apiResponse = await response.Content.ReadAsStringAsync();

service = JsonConvert.DeserializeObject<Service>(apiResponse);

}

}

return View(service);

}

[HttpPost]

public async Task<IActionResult> UpdateService(Service service, int id)

{

Service receivedservice = new Service();

using (var httpClient = new HttpClient())

{

StringContent content = new StringContent(JsonConvert.SerializeObject(service), Encoding.UTF8, "application/json");

using (var response = await httpClient.PutAsync("https://localhost:7015/api/Services/" + id, content))

{

string apiResponse = await response.Content.ReadAsStringAsync();

ViewBag.Result = "Success";

receivedservice = JsonConvert.DeserializeObject<Service>(apiResponse);

System.Diagnostics.Debug.WriteLine($"Response JSON: {apiResponse}");

}

}

return View(receivedservice);

}

[HttpPost]

public async Task<IActionResult> DeleteService(int IdService)

{

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.DeleteAsync("https://localhost:7015/api/Services/" + IdService))

{

string apiResponse = await response.Content.ReadAsStringAsync();

}

}

return RedirectToAction("AdminService");

}

//Вывод дат Админ

public async Task<IActionResult> AdminDateAppointment()

{

List<DateAppointment> dateAppList = new List<DateAppointment>();

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.GetAsync("https://localhost:7015/api/DateAppointments"))

{

string apiResponse = await response.Content.ReadAsStringAsync();

dateAppList = JsonConvert.DeserializeObject<List<DateAppointment>>(apiResponse);

}

}

return View(dateAppList);

}

public ViewResult AddDateAppointment() => View();

[HttpPost]

public async Task<IActionResult> AddDateAppointment(DateAppointment dateAppointment)

{

DateAppointment reciveddateAppointment = new DateAppointment();

using (var httpClient = new HttpClient())

{

StringContent content = new StringContent(JsonConvert.SerializeObject(dateAppointment), Encoding.UTF8, "application/json");

using (var response = await httpClient.PostAsync("https://localhost:7015/api/DateAppointments", content))

{

string apiResponse = await response.Content.ReadAsStringAsync();

ViewBag.Result = "Success";

reciveddateAppointment = JsonConvert.DeserializeObject<DateAppointment>(apiResponse);

System.Diagnostics.Debug.WriteLine($"Response JSON: {apiResponse}");

}

}

return View(reciveddateAppointment);

}

public async Task<IActionResult> UpdateDateAppointment(int id)

{

DateAppointment dateAppointment = new DateAppointment();

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.GetAsync("https://localhost:7015/api/DateAppointments/" + id))

{

string apiResponse = await response.Content.ReadAsStringAsync();

dateAppointment = JsonConvert.DeserializeObject<DateAppointment>(apiResponse);

}

}

return View(dateAppointment);

}

[HttpPost]

public async Task<IActionResult> UpdateDateAppointment(DateAppointment dateAppointment, int id)

{

DateAppointment receiveddateAppointment = new DateAppointment();

using (var httpClient = new HttpClient())

{

StringContent content = new StringContent(JsonConvert.SerializeObject(dateAppointment), Encoding.UTF8, "application/json");

using (var response = await httpClient.PutAsync("https://localhost:7015/api/DateAppointments/" + id, content))

{

string apiResponse = await response.Content.ReadAsStringAsync();

ViewBag.Result = "Success";

receiveddateAppointment = JsonConvert.DeserializeObject<DateAppointment>(apiResponse);

System.Diagnostics.Debug.WriteLine($"Response JSON: {apiResponse}");

}

}

return View(receiveddateAppointment);

}

[HttpPost]

public async Task<IActionResult> DeleteDateAppointment(int IdDateAppointment)

{

using (var httpClient = new HttpClient())

{

using (var response = await httpClient.DeleteAsync("https://localhost:7015/api/DateAppointments/" + IdDateAppointment))

{

string apiResponse = await response.Content.ReadAsStringAsync();

}

}

return RedirectToAction("AdminDateAppointment");

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}

}

1. AddAdminAppointment

@model DentistryWeb.Models.DateAppointment

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/main-doctor.css">

<title>Добавление даты приёма</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("AdminService")">Услуги</a></li>

<li><a href="@Url.Action("AdminDateAppointment")">Даты приёма</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="history">

<table>

<thead>

<tr>

<th>Дата</th>

<th>Время</th>

<th>Сохранить</th>

</tr>

</thead>

<tbody>

<tr>

<form method="post" action="@Url.Action("AddDateAppointment", "Home")">

<td>

<input asp-for="Date" type="date" />

</td>

<td>

<input asp-for="Time" type="time" />

</td>

<td>

<button>Сохранить</button>

</td>

<input type="hidden" name="Free" value="True" />

</form>

</tr>

</tbody>

</table>

</div>

@if (ViewBag.Result == "Success")

{

<p> Дата успешно добавлена</p>

}

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

10. AddService

@model DentistryWeb.Models.Service

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/main-doctor.css">

<title>Добавление услуги</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("AdminService")">Услуги</a></li>

<li><a href="@Url.Action("AdminDateAppointment")">Даты приёма</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="history">

<table>

<thead>

<tr>

<th>Название</th>

<th>Цена, р</th>

<th>Сохранить</th>

</tr>

</thead>

<tbody>

<tr>

<form method="post">

<td>

<input asp-for="NameService" />

</td>

<td>

<input asp-for="PriceService" />

</td>

<td>

<button>Сохранить</button>

</td>

</form>

</tr>

</tbody>

</table>

</div>

@if (ViewBag.Result == "Success")

{

<p> Услуга успешно добавлена</p>

}

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. AdminDateAppointment

@model IEnumerable<DentistryWeb.Models.DateAppointment>

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/main-doctor.css">

<title>Редактор дат приёмов</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("AdminService")">Услуги</a></li>

<li><a href="@Url.Action("AdminDateAppointment")">Даты приёма</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="history">

<table>

<thead>

<tr>

<th>Дата</th>

<th>Время</th>

<th>Доступность</th>

<th>Изменить</th>

<th>Удалить</th>

</tr>

</thead>

<tbody>

@foreach (var r in Model)

{

@if (@r.Free)

{

<tr>

<td>@r.Date.ToString("dd-MM-yyyy")</td>

<td>@r.Time.ToString(@"hh\:mm")</td>

<td>Свободно</td>

<td>

<a asp-action="UpdateDateAppointment" asp-route-id="@r.IdDateAppointment">

<button>Изменить</button>

</a>

</td>

<td>

<form asp-action="DeleteDateAppointment" method="post">

<input type="hidden" value="@r.IdDateAppointment" name="IdDateAppointment" />

<button>Удалить</button>

</form>

</td>

</tr>

}

else{

<tr>

<td>@r.Date.ToString("dd-MM-yyyy")</td>

<td>@r.Time.ToString(@"hh\:mm")</td>

<td>Занято</td>

<td>Изменение недоступно</td>

<td>Удаление недоступно</td>

</tr>

}

}

</tbody>

</table>

</div>

<div class="add-button">

<a asp-action="AddDateAppointment">

<button style="width: 200px;">Добавить дату приёма</button>

</a>

</div>

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. AdminService

@model IEnumerable<DentistryWeb.Models.Service>

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/main-doctor.css">

<title>Главная страница</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("AdminService")">Услуги</a></li>

<li><a href="@Url.Action("AdminDateAppointment")">Даты приёма</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="history">

<table>

<thead>

<tr>

<th>Название</th>

<th>Цена</th>

<th>Изменить</th>

<th>Удалить</th>

</tr>

</thead>

<tbody>

@foreach (var r in Model)

{

<tr>

<td>@r.NameService</td>

<td>@r.PriceService р</td>

<td>

<a asp-action="UpdateService" asp-route-id="@r.IdService">

<button>Изменить</button>

</a>

</td>

<td>

<form asp-action="DeleteService" method="post">

<input type="hidden" value="@r.IdService" name="IdService" />

<button>Удалить</button>

</form>

</td>

</tr>

}

</tbody>

</table>

</div>

<div class ="add-button">

<a asp-action="AddService">

<button style="width: 200px;">Добавить услугу</button>

</a>

</div>

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

12. Auth

@model DentistryWeb.Models.Patient

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/auth.css">

<title>Авторизация</title>

</head>

<body>

<section>

<div class="form-box">

<div class="form-value">

<form asp-controller="Home" asp-action="Auth" method="post">

<h2>Авторизация</h2>

<div class="inputbox">

<ion-icon name="mail"></ion-icon>

<input asp-for="EmailPatient" type="email" required>

<label asp-for="EmailPatient">Почта</label>

</div>

<div class="inputbox">

<ion-icon name="lock"></ion-icon>

<input asp-for="PasswordPatient" type="password" required>

<label asp-for="PasswordPatient">Пароль</label>

</div>

<button type="submit">Войти</button>

<div class="register">

<p>Нет аккаунта? <a href="@Url.Action("Reg")">Регистрация</a></p>

@if (ViewBag.ErrorMessage != null)

{

<div class="error-message">

@ViewBag.ErrorMessage

</div>

}

</div>

</form>

</div>

</div>

</section>

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. CreateApp

@model Tuple<List<Service>, List<Employee>, List<DateAppointment>>

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/app.css">

<title>Запись на приём</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("MainPatient")">Главная</a></li>

<li><a href="@Url.Action("Service")">Услуги</a></li>

<li><a href="@Url.Action("ProfilePatient")">Личный кабинет</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="message">

@if (TempData["SuccessMessage"] != null)

{

<div class="alert alert-success">

@TempData["SuccessMessage"]

</div>

}

</div>

<div class="sel">

<form method="post" action="@Url.Action("CreateApp", "Home")">

<p>Выберете услугу: </p>

<select name="ServiceId">

@foreach (var service in Model.Item1)

{

<option value="@service.IdService">@service.NameService</option>

}

</select>

<p>Выберете врача: </p>

<select name="EmployeeId">

@foreach (var emp in Model.Item2)

{

@if(emp.RoleId == 2)

{

<option value="@emp.IdEmployee">@emp.SurnameEmployee @emp.FirstNameEmployee @emp.MiddleEmployee</option>

}

}

</select>

<p>Выберете дату приёма: </p>

<select name="DateAppointmentId">

@foreach (var date in Model.Item3)

{

@if (date.Free == true)

{

<option value="@date.IdDateAppointment">@date.Date.ToString("dd-MM-yyyy") @date.Time.ToString(@"hh\:mm")</option>

}

}

</select>

<input type="hidden" name="Number" value="@GenerateRandomNumber()" />

<input type="hidden" name="Complaint" value="Не заполнено" />

<input type="hidden" name="Treatment" value="Не заполнено" />

<input type="hidden" name="PatientId" value="1" />

<div class="create">

<button>Подтвердить запись</button>

</div>

</form>

</div>

@functions {

public string GenerateRandomNumber()

{

Random rand = new Random();

return rand.Next(1000, 9999).ToString();

}

}

<div class="image">

<img src="~/image/app.jpg" alt="pic" width="650" height="540">

</div>

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. MainDoctor

@model Tuple<List<Appointment>, List<Patient>, List<DateAppointment>>

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/main-doctor.css">

<title>Главная страница</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("MainDoctor")">Истории болезней</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="history">

<table>

<thead>

<tr>

<th>Жалоба</th>

<th>Лечение</th>

<th>Фамилия пациента</th>

<th>Имя пациента</th>

<th>Отчество пациента</th>

<th>Дата приёма</th>

<th>Изменить</th>

<th>Удалить</th>

</tr>

</thead>

<tbody>

@foreach (var appointment in Model.Item1)

{

<tr>

<td>@appointment.Complaint</td>

<td>@appointment.Treatment</td>

@{

var patient = Model.Item2.FirstOrDefault(p => p.IdPatient == appointment.PatientId);

var surnamePatient = (patient != null) ? patient.SurnamePatient : "";

var firstNamePatient = (patient != null) ? patient.FirstNamePatient : "";

var middlePatient = (patient != null) ? patient.MiddlePatient : "";

}

<td>@surnamePatient</td>

<td>@firstNamePatient</td>

<td>@middlePatient</td>

@{

var dateApp = Model.Item3.FirstOrDefault(p => p.IdDateAppointment == appointment.DateAppointmentId);

var date = (dateApp != null) ? dateApp.Date : DateTime.MinValue; ;

var time = (dateApp != null) ? dateApp.Time : TimeSpan.Zero;

}

<td>@date.ToString("dd-MM-yyyy") @time.ToString(@"hh\:mm")</td>

<td>

<a asp-action="UpdateAppointment" asp-route-id="@appointment.IdAppointment">

<button>Изменить</button>

</a>

</td>

<td>

<form asp-action="DeleteAppointment" method="post">

<input type="hidden" value="@appointment.IdAppointment" name="IdAppointment" />

<button>Удалить</button>

</form>

</td>

</tr>

}

</tbody>

</table>

</div>

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. MainPatient

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/main-patient.css">

<title>Главная страница</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("MainPatient")">Главная</a></li>

<li><a href="@Url.Action("Service")">Услуги</a></li>

<li><a href="@Url.Action("ProfilePatient")">Личный кабинет</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="name">

<h1>Healthy smile</h1>

<p>Настоящие улыбки начинаются здесь!</p>

<div class="create">

<a asp-action="CreateApp">

<button>Записаться на приём</button>

</a>

</div>

</div>

<div class="image">

<img src="~/image/main\_image.jpg" alt="logo" width="500" height="500">

</div>

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. ProfilePatient

@model Tuple<List<Appointment>, List<Service>, List<Employee>, List<DateAppointment>>

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/main-doctor.css">

<title>Главная страница</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("MainPatient")">Главная</a></li>

<li><a href="@Url.Action("Service")">Услуги</a></li>

<li><a href="@Url.Action("ProfilePatient")">Личный кабинет</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="history">

<table>

<thead>

<tr>

<th>Номер талона</th>

<th>Услуга</th>

<th>Стоимость</th>

<th>Врач</th>

<th>Дата приёма</th>

<th>Отменить</th>

</tr>

</thead>

<tbody>

@foreach (var appointment in Model.Item1)

{

<tr>

<td>@appointment.Number</td>

@{

var service = Model.Item2.FirstOrDefault(p => p.IdService == appointment.ServiceId);

var nameService = (service != null) ? service.NameService : "";

int priceService = (service != null) ? service.PriceService : 0;

}

<td>@nameService</td>

<td>@priceService</td>

@{

var emp = Model.Item3.FirstOrDefault(p => p.IdEmployee == appointment.EmployeeId);

var surName = (emp != null) ? emp.SurnameEmployee : "";

var firstName = (emp != null) ? emp.FirstNameEmployee : "";

var middleName = (emp != null) ? emp.MiddleEmployee : "";

}

<td>@surName @firstName @middleName</td>

@{

var dateApp = Model.Item4.FirstOrDefault(p => p.IdDateAppointment == appointment.DateAppointmentId);

var date = (dateApp != null) ? dateApp.Date : DateTime.MinValue; ;

var time = (dateApp != null) ? dateApp.Time : TimeSpan.Zero;

}

<td>@date.ToString("dd-MM-yyyy") @time.ToString(@"hh\:mm")</td>

<td>

<form asp-action="DeleteAppointmentPatient" method="post">

<input type="hidden" value="@appointment.IdAppointment" name="IdAppointment" />

<button>Отменить запись</button>

</form>

</td>

</tr>

}

</tbody>

</table>

</div>

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. Reg

@model DentistryWeb.Models.Patient

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/reg.css">

<title>Регистрация</title>

</head>

<body>

<section>

<div class="form-box">

<div class="form-value">

<form asp-controller="Home" asp-action="Reg" method="post">

<h2>Регистрация</h2>

<div class="inputbox">

<ion-icon name="mail"></ion-icon>

<input asp-for="EmailPatient" type="email" required>

<label asp-for="EmailPatient">Почта</label>

</div>

<div class="inputbox">

<ion-icon name="lock"></ion-icon>

<input asp-for="PasswordPatient" type="password" required>

<label asp-for="PasswordPatient">Пароль</label>

</div>

<div class="inputbox">

<ion-icon name="lock"></ion-icon>

<input asp-for="SurnamePatient" type="text" required>

<label asp-for="SurnamePatient">Фамилия</label>

</div>

<div class="inputbox">

<ion-icon name="lock"></ion-icon>

<input asp-for="FirstNamePatient" type="text" required>

<label asp-for="FirstNamePatient">Имя</label>

</div>

<div class="inputbox">

<ion-icon name="lock"></ion-icon>

<input asp-for="MiddlePatient" type="text" required>

<label asp-for="MiddlePatient">Отчество</label>

</div>

<div class="inputbox">

<ion-icon name="lock"></ion-icon>

<input asp-for="Age" required>

<label asp-for="Age">Возраст</label>

</div>

<select name="Sex">

<option value="Sex">Мужчина</option>

<option value="Sex">Женщина</option>

<option value="Sex">Другое</option>

</select>

<input type="hidden" name="RoleId" value="1" />

<button type="submit">Зарегистрироваться</button>

<div class="register">

<p>Передумали? <a href="@Url.Action("Auth")">Авторизация</a></p>

@if (ViewBag.ErrorMessage != null)

{

<div class="error-message">

@ViewBag.ErrorMessage

</div>

}

</div>

</form>

</div>

</div>

</section>

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. Service

@model IEnumerable<DentistryWeb.Models.Service>

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/service.css">

<title>Услуги</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("MainPatient")">Главная</a></li>

<li><a href="@Url.Action("Service")">Услуги</a></li>

<li><a href="@Url.Action("ProfilePatient")">Личный кабинет</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="service">

@foreach (var r in Model)

{

<button>@r.NameService <br>Цена: @r.PriceService р</button>

}

</div>

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. UpdateAppointment

@model DentistryWeb.Models.Appointment

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/main-doctor.css">

<title>Изменить историю болезни</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("MainDoctor")">Истории болезней</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

@if (ViewBag.Result == "Success")

{

<p> История болезни изменена</p>

}

<div class="history">

<table >

<thead>

<tr>

<th>Жалоба</th>

<th>Лечение</th>

<th>Сохранить изменения</th>

</tr>

</thead>

<tbody>

<tr>

<form method="post">

<input style="display: none" asp-for="IdAppointment" readonly />

<input style="display: none" asp-for="Number" readonly />

<input style="display: none" asp-for="EmployeeId" readonly />

<input style="display: none" asp-for="PatientId" readonly />

<input style="display: none" asp-for="ServiceId" readonly />

<input style="display: none" asp-for="DateAppointmentId" readonly />

<td>

<input style=" width: 100%;padding: 5px;box-sizing: border-box;" asp-for="Complaint" />

</td>

<td>

<input style=" width: 100%;padding: 5px;box-sizing: border-box;" asp-for="Treatment" />

</td>

<td>

<button>Сохранить</button>

</td>

</form>

</tr>

</tbody>

</table>

</div>

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. UpdateDateAppointment

@model DentistryWeb.Models.DateAppointment

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/main-doctor.css">

<title>Изменение даты приёма</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("AdminService")">Услуги</a></li>

<li><a href="@Url.Action("AdminDateAppointment")">Даты приёма</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="history">

<table>

<thead>

<tr>

<th>Дата</th>

<th>Время</th>

<th>Сохранить</th>

</tr>

</thead>

<tbody>

<tr>

<form method="post" action="@Url.Action("UpdateDateAppointment", "Home")">

<td>

<input asp-for="Date" type="date" />

</td>

<td>

<input asp-for="Time" type="time" />

</td>

<td>

<button>Сохранить</button>

</td>

<input type="hidden" name="Free" value="True" />

</form>

</tr>

</tbody>

</table>

</div>

@if (ViewBag.Result == "Success")

{

<p> Дата успешно изменена</p>

}

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>

1. UpdateService

@model DentistryWeb.Models.Service

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="~/css/main-doctor.css">

<title>Изменить услугу</title>

</head>

<body>

<header>

<ul>

<li><ion-icon name="pin"></ion-icon><a href="">г. Москва, ул. Родниковая, д.7</a></li>

<li><ion-icon name="time"></ion-icon><a href="">Часы работы: 09:00-21:00</a></li>

<li><ion-icon name="call"></ion-icon><a href="">Телефон: +7 (495) 885-80-80</a></li>

<li><ion-icon name="search"></ion-icon><a href="">Поиск:</a></li>

</ul>

</header>

<nav>

<div class="logo-container">

<img src="~/image/logo.png" alt="logo" width="80" height="80">

</div>

<ul>

<li><a href="@Url.Action("AdminService")">Услуги</a></li>

<li><a href="@Url.Action("AdminDateAppointment")">Даты приёма</a></li>

</ul>

<div class="exit">

<a href="@Url.Action("Auth")">Выход</a>

</div>

</nav>

<div class="history">

<table>

<thead>

<tr>

<th>Название</th>

<th>Цена, р</th>

<th>Сохранить изменения</th>

</tr>

</thead>

<tbody>

<tr>

<form method="post">

<input hidden asp-for="IdService" readonly />

<td>

<input asp-for="NameService" />

</td>

<td>

<input asp-for="PriceService" />

</td>

<td>

<button>Сохранить</button>

</td>

</form>

</tr>

</tbody>

</table>

</div>

@if (ViewBag.Result == "Success")

{

<div class="message">

<p> Услуга изменена</p>

</div>

}

<script src="https://unpkg.com/ionicons@4.1.2/dist/ionicons.js"></script>

</body>

</html>