**Приложение В**

(информационное)

**Листинги программного продукта**

**Листинг В.1 – Отображение меню**

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container">

<img src="~/logo.svg" width="60" height="60" class="d-inline-block align-top">

<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Главная</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 justify-content-between">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="User" asp-action="Index">Пользователи</a></li>

<li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="Role" asp-action="Index">Роли</a></li>

<li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="City" asp-action="Index">Города</a></li>

<li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="Attraction" asp-action="Index">Достопримечательности</a></li>

<li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="DataReference" asp-action="Index">Связи</a></li>

<li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="GameScore" asp-action="Index">Игры</a></li>

<li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="TestQuestion" asp-action="Index">Вопросы</a></li>

<li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="Account" asp-action="Logout">Выход</a></li>

</ul>

</div>

</div>

</nav>

**Листинг В.2 – Отображение лучших результатов игр**

private async Task LoadData() {

try

{

Game1Result.IsVisible = true;

Game1NoResult.IsVisible = false;

game1Score = await \_gameScoreService.Get(1);

Game1UserName.Text = game1Score.UserName;

Game1UserScore.Text = $"Время: {game1Score.Score}";

if (game1Score.Image == null) {

Game1UserImage.Source = "icon\_no\_avatar.jpg";

}

else

{

Game1UserImage.Source = $"{AppSettings.Url}images/Users/{game1Score.UserId}/{game1Score.Image}"; }

}

catch

{

Game1Result.IsVisible = false;

Game1NoResult.IsVisible = true;

}

try

{

Game2Result.IsVisible = true;

Game2NoResult.IsVisible = false;

game2Score = await \_gameScoreService.Get(2);

Game2UserName.Text = game2Score.UserName;

Game2UserScore.Text = $"Правильных ответов: {game2Score.Score}";

if (game2Score.Image == null) {

Game2UserImage.Source = "icon\_no\_avatar.jpg";

}

else {

Game2UserImage.Source = $"{AppSettings.Url}images/Users/{game2Score.UserId}/{game2Score.Image}";

}

}

catch {

Game2Result.IsVisible = false;

Game2NoResult.IsVisible = true;

}

}

**Листинг В.3 – Тестирование изменения существующего пользователя**

[Fact]

public void Edit\_ReturnsView()

{

var result = \_controller.Edit(2);

Assert.IsType<ViewResult>(result);

}

[Fact]

public void Edit\_InvalidModelState\_ReturnsView()

{

\_controller.ModelState.AddModelError("Email", "Email уже используется");

var newObj = new User

{

UserId = 2, UserName = "TestName3", Email = "test1@mail.com", Password = "testPassword3", RoleId = 1

};

var result = \_controller.Edit(newObj, null);

var viewResult = Assert.IsType<ViewResult>(result);

var testEmployee = Assert.IsType<User>(viewResult.Model);

Assert.Equal(newObj.UserName, testEmployee.UserName);

Assert.Equal(newObj.Email, testEmployee.Email);

}

[Fact]

public void Edit\_InvalidId\_ReturnsNotFoundResult()

{

var result = \_controller.Edit(-1);

Assert.IsType<NotFoundResult>(result);

}

[Fact]

public void Edit\_RedirectsToIndex()

{

var newObj = new User

{

UserId = 2, UserName = "TestName3", Email = "test3@mail.com", Password = "testPassword3", RoleId = 1

};

var result = \_controller.Edit(newObj, null);

var redirectToActionResult = Assert.IsType<RedirectToActionResult>(result);

Assert.Equal("Index", redirectToActionResult.ActionName);

}