Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Лабораторная работа 2

по дисциплине «Технологии разработки WEB-приложений»

«Знакомство с технологией ASP.NET MVC»

|  |  |  |
| --- | --- | --- |
| Студент |  | Ковальчук А. Д. |
| Группа | АИ-14 |  |
| Руководитель |  | Лаврухина Т. В. |

Липецк – 2018 г.

Цель работы

Познакомиться с основами технологии ASP.NET MVC. На практике создать базовую функциональность сайта через паттерн Model-View-Controller.

Задание

Используя созданный в первой лабораторной работе шаблон, создать проект ASP.NET MVC.

К работе предъявлены следующие требования:

1. Верстка должна быть разбита на серию View и один Layout.
2. Исчисляемый контент должен выводиться на View при помощи конструкций языка C#.
3. Исчисляемый контент должен передаваться во View через контроллер в качестве модели.
4. Переходы между страницами должны быть реализованы при помощи контроллеров.

# Результаты выполнения работы

## Описание предметной области

В качестве предметной области выбран Web-сервис для ведения и блогов и чтения статей.

## Коды страниц

### Коды контроллеров

HomeController

namespace Lab1.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

return View();

}

}

}

BlogsController

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Data.Entity.Migrations;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using Lab1.Models;

namespace Lab1.Controllers

{

public class BlogsController : Controller

{

const int CAFP = 10; // COUNT\_ARTICLES\_FOR\_PAGE - количество статей на странице

ArticleContext db = new ArticleContext();

[HttpGet]

public ActionResult Index(int? CategoryID, int? page, int? message)

{

switch (message)

{

case 1:

ViewBag.Message = "Статья удалена!";

break;

case 2:

ViewBag.Message = "Статья, которую вы пытаетесь удалить, не существует.";

ViewBag.MessageCode = 1;

break;

case 3:

ViewBag.Message = "Статья успешно добавлена!";

break;

case 4:

ViewBag.Message = "Статья, которую вы пытались отредактировать, не существует.";

ViewBag.MessageCode = 1;

break;

case 5:

ViewBag.Message = "Изменения успешно зафиксированы!";

break;

case 6:

ViewBag.Message = "Статья уже находится у вас в Избранном.";

ViewBag.MessageCode = 1;

break;

case 7:

ViewBag.Message = "Статья успешно добавлена в избранное!";

break;

case 8:

ViewBag.Message = "Не найден пользователь или статья для добавления ее в Избранное.";

ViewBag.MessageCode = 1;

break;

case 9:

ViewBag.Message = "Не найден пользователь и статья для удаления ее из Избранного.";

ViewBag.MessageCode = 1;

break;

case 10:

ViewBag.Message = "Не определен текущий пользователь, чтобы открыть его Избранные статьи.";

ViewBag.MessageCode = 1;

break;

}

ViewBag.Categorys = db.Categorys;

ViewBag.Favorites = db.Favorites;

ViewBag.categoryID = 0;

if (CategoryID != null)

{

ViewBag.categoryID = CategoryID;

ViewBag.Authors = db.Users;

List<Article> ArticleForLooking = db.Articles

.Where(x => x.CategoryID == CategoryID)

.ToList();

ArticleForLooking.Reverse();

ViewBag.CountPage = (int)Math.Ceiling((double)ArticleForLooking.Count / CAFP);

ViewBag.FirstPageDiasbled = "";

ViewBag.LastPageDiasbled = "";

// Если статей меньше CAFP в этой категории, то выводим все

if (page == null)

{

// Если больше CAFP, но пользователь не указывал какую страницу открыть, то

// открываем превую

if (ViewBag.CountPage > 1)

{

ArticleForLooking = ArticleForLooking.GetRange(0, CAFP);

ViewBag.CurrentPage = 1; //В этом случае текущей страницей будет первая

ViewBag.FirstPageDiasbled = "disabled"; //Выключаем возврат на предыдующую страницу

}

}

else

{

// Если статей больше CAFP в этой категории, то выводим CAFP статей для указанной страницы

//Если на последней странице нет CAFP статей

if (CAFP \* (int)page >= ArticleForLooking.Count)

{

ArticleForLooking = ArticleForLooking.GetRange(CAFP \* ((int)page - 1), ArticleForLooking.Count - CAFP \* ((int)page - 1));

ViewBag.LastPageDiasbled = "disabled"; //Выключаем переход на следующую страницу

}

else

{

//Если на странице есть CAFP статей, то берем их

ArticleForLooking = ArticleForLooking.GetRange(CAFP \* ((int)page - 1), CAFP);

}

ViewBag.CurrentPage = page; //Получаем тукущую страницу

if (page == 1) ViewBag.FirstPageDiasbled = "disabled";

}

return View(ArticleForLooking);

}

return View();

}

public ActionResult Favorites(int? CurrentUserID, int? page, int? message)

{

if (CurrentUserID == null) return RedirectToAction("Index", "Blogs", new { message = 10 });

switch (message)

{

case 1:

ViewBag.Message = "Статья успешно удалена из вашего избранного!";

break;

case 2:

ViewBag.Message = "У вас в Избранном нет статьи, которую вы пытаетесь удалить.";

ViewBag.MessageCode = 1;

break;

case 3:

ViewBag.Message = "Не найдена статья для удаления ее из Избранного.";

ViewBag.MessageCode = 1;

break;

case 4:

ViewBag.Message = "Статья удалена!";

break;

case 5:

ViewBag.Message = "Статья, которую вы пытаетесь удалить, не существует.";

ViewBag.MessageCode = 1;

break;

}

ViewBag.Categorys = db.Categorys;

ViewBag.Articles = db.Articles;

ViewBag.Authors = db.Users;

List<Favorites> favorites = db.Favorites.Where(x => x.UserID == CurrentUserID).ToList();

favorites.Reverse();

ViewBag.CountPage = (int)Math.Ceiling((double)favorites.Count / CAFP);

ViewBag.FirstPageDiasbled = "";

ViewBag.LastPageDiasbled = "";

// Если статей меньше CAFP в Избранном, то выводим все

if (page == null)

{

// Если больше CAFP, но пользователь не указывал какую страницу открыть, то

// открываем превую

if (ViewBag.CountPage > 1)

{

favorites = favorites.GetRange(0, CAFP);

ViewBag.CurrentPage = 1; //В этом случае текущей страницей будет первая

ViewBag.FirstPageDiasbled = "disabled"; //Выключаем возврат на предыдующую страницу

}

}

else

{

// Если статей больше CAFP в Избранном, то выводим CAFP статей для указанной страницы

//Если на последней странице нет CAFP статей

if (CAFP \* (int)page >= favorites.Count)

{

favorites = favorites.GetRange(CAFP \* ((int)page - 1), favorites.Count - CAFP \* ((int)page - 1));

ViewBag.LastPageDiasbled = "disabled"; //Выключаем переход на следующую страницу

}

else

{

//Если на странице есть CAFP статей, то берем их

favorites = favorites.GetRange(CAFP \* ((int)page - 1), CAFP);

}

ViewBag.CurrentPage = page; //Получаем тукущую страницу

if (page == 1) ViewBag.FirstPageDiasbled = "disabled";

}

return View(favorites);

}

//Метод добавления статьи в избранное

public ActionResult AddFavorites(int? CurrentUserID, int? ArticleID, int? page)

{

int message = 8;

if (db.Users.Find(CurrentUserID) != null && db.Articles.Find(ArticleID) != null)

{

message = 6;

if (db.Favorites.Where(x => x.UserID == CurrentUserID && x.ArticleID == ArticleID).ToList().Count() == 0)

{

db.Favorites.Add(new Favorites { UserID = (int)CurrentUserID, ArticleID = (int)ArticleID });

db.SaveChanges();

message = 7;

}

return new RedirectResult(Url.Action("Index", "Blogs", new { db.Articles.Find(ArticleID).CategoryID, page, message }) + "#article" + ArticleID);

}

else

{

return RedirectToAction("Index", "Blogs", new { message });

}

}

//Метод удаления статьи из избранного

public ActionResult DelFavorites(int? CurrentUserID, int? ArticleID, int? page, int? NextArticleID)

{

int message = 0;

if (db.Users.Find(CurrentUserID) != null && db.Articles.Find(ArticleID) != null)

{

message = 2;

if (db.Favorites.Where(x => x.UserID == CurrentUserID && x.ArticleID == ArticleID).ToList().Count > 0)

{

Favorites fav = db.Favorites.Where(x => x.UserID == CurrentUserID && x.ArticleID == ArticleID).ToList()[0];

db.Favorites.Remove(fav);

db.SaveChanges();

message = 1;

}

return new RedirectResult(Url.Action("Favorites", "Blogs", new { CurrentUserID, page, message }) + "#article" + NextArticleID);

}

else

{

message = 3;

if (db.Users.Find(CurrentUserID) != null)

{

return new RedirectResult(Url.Action("Favorites", "Blogs", new { CurrentUserID, page, message }) + "#article" + NextArticleID);

}

else

{

message = 9;

return new RedirectResult(Url.Action("Index", "Blogs", new { page, message }) + "#article" + NextArticleID);

}

}

}

//Метод удаления статьи

public ActionResult Delete(int? CurrentUserID, int? CategoryID, int? ArticleID, int? page, int? NextArticleID)

{

Article art = db.Articles.Find(ArticleID);

if (art != null)

{

List<Favorites> favorites = db.Favorites.Where(x => x.ArticleID == ArticleID).ToList();

if (favorites.Count > 0)

{

foreach (Favorites fav in favorites)

{

db.Favorites.Remove(fav);

}

}

int catID = art.CategoryID;

db.Articles.Remove(art);

db.SaveChanges();

if (CurrentUserID == null)

{

return new RedirectResult(Url.Action("Index", "Blogs", new { CategoryID = catID, page, message = 1 }) + "#article" + NextArticleID);

}

else

{

return new RedirectResult(Url.Action("Favorites", "Blogs", new { CurrentUserID, page, message = 4 }) + "#article" + NextArticleID);

}

}

if (CurrentUserID == null)

{

return new RedirectResult(Url.Action("Index", "Blogs", new { CategoryID, page, message = 2 }) + "#article" + NextArticleID);

}

else

{

return new RedirectResult(Url.Action("Favorites", "Blogs", new { CurrentUserID, page, message = 5 }) + "#article" + NextArticleID);

}

}

//Метод отображения страницы создания статьи

public ActionResult Create(int? CategoryID, string Subject, int? message)

{

if (message == 1) ViewBag.Message = "Все поля обязательны для заполнения";

ViewBag.Subject = Subject;

List<Category> CategoryForLooking = db.Categorys.ToList();

return View(CategoryForLooking);

}

//Метод создания статьи

[HttpPost]

public ActionResult Create(Article art)

{

if (art.Subject != null && art.Subject != "" && art.Text != null && art.Text != "" && art.CategoryID != 0)

{

db.Articles.Add(art);

db.SaveChanges();

return RedirectToAction("Index", "Blogs", new { art.CategoryID, message = 3 });

}

return RedirectToAction("Create", "Blogs", new { art.CategoryID, art.Subject, message = 1 });

}

//Метод отображения страницы редактирования статьи

public ActionResult Edit(int? ArticleID, string Subject, int? message, int? page, int? CategoryID)

{

if (message == 1) ViewBag.Message = "Все поля обязательны для заполнения";

ViewBag.Subject = Subject;

Article art = db.Articles.Find(ArticleID);

ViewBag.Categorys = db.Categorys.ToList();

if (art == null) return new RedirectResult(Url.Action("Index", "Blogs", new { CategoryID, page, message = 4 }) + "#article" + ArticleID);

return View(art);

}

//Метод редактирования статьи

[HttpPost]

public ActionResult Edit(Article art)

{

if (art.Subject != null && art.Subject != "" && art.Text != null && art.Text != "" && art.CategoryID != 0)

{

db.Articles.AddOrUpdate(art);

db.SaveChanges();

//считаем какая будет новая страница для отредактированной статьи

List<Article> articles = db.Articles.Where(x => x.CategoryID == art.CategoryID).ToList();

articles.Reverse();

int number = 0;

for (int i=0; i<articles.Count(); i++)

{

if(articles[i].ID == art.ID)

{

number = i + 1;

break;

}

}

int page = (int)Math.Ceiling((double)number / CAFP);

return new RedirectResult(Url.Action("Index", "Blogs", new { art.CategoryID, page, message = 5 }) + "#article" + art.ID);

}

return RedirectToAction("Edit", "Blogs", new { ArticleID = art.ID, art.Subject, message = 1 });

}

}

}

RegistrationController

namespace Lab1.Controllers

{

public class RegistrationController : Controller

{

// GET: Registration

public ActionResult Index()

{

return View();

}

}

}

### Код шаблона с навигационной панелью и футером

<!doctype html>

<html lang="ru">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>MyBlog</title>

@Styles.Render("~/Content/css")

</head>

<body>

@\*Навигационная панель\*@

<nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">

<span class="navbar-brand">

<img src="~/Content/Images/apple-touch-icon.png" width="30" height="30" class="d-inline-block align-top" alt="">

MyBlog

</span>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarTogglerDemo02">

<ul class="navbar-nav mr-auto mt-2 mt-lg-0">

<li class="@ViewBag.Page1">

@Html.ActionLink("Главная", "Index", "Home", new { area = "" }, new { @class = "nav-link" })

</li>

<li class="@ViewBag.Page2">

@Html.ActionLink("Блоги", "Index", "Blogs", new { area = "" }, new { @class = "nav-link" })

</li>

</ul>

<form class="form-inline" @ViewBag.Hidden>

<div class="input-group">

<input type="email" class="form-control form-control-sm mr-sm-2" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Email">

</div>

<div class="input-group">

<input type="password" class="form-control form-control-sm mr-sm-2 my-2 my-lg-0" id="exampleInputPassword1" placeholder="Пароль">

</div>

<div class="input-group">

<button type="button" class="btn btn-outline-success btn-sm mr-2">Вход</button>

<button type="button" class="btn btn-outline-warning btn-sm" onclick="location.href='@Url.Action("Index", "Registration")'">Регистрация</button>

</div>

</form>

<ul class="navbar-nav ml-auto mt-2 mt-lg-0" @ViewBag.Hidden1>

<li class="@ViewBag.Page3">

@\*Передаем ID текущего пользователя для отображения его Избранных статей (без регистрации и авторизации задаем по умолчанию id=1)\*@

@Html.ActionLink("Избранное", "Favorites", "Blogs", new { CurrentUserID = 1 }, new { @class = "nav-link" })

</li>

<li class="">

@Html.ActionLink("Антон Ковальчук", "", "", new { area = "" }, new { @class = "nav-link" })

</li>

</ul>

</div>

</nav>

@\*Основной контент с других страниц\*@

<div class="container body-content">

@RenderBody()

<hr />

<footer>

<p>&copy; Ковальчук А. Д., @DateTime.Now.Year – MyBlog</p>

</footer>

</div>

@Scripts.Render("~/bundles/jquery")

@Scripts.Render("~/bundles/popper")

@Scripts.Render("~/bundles/bootstrap")

@RenderSection("scripts", required: false)

</body>

</html>

### Коды некоторых файлов представлений

Home/Index.cshtml

@{

ViewBag.Page1 = "nav-item active";

ViewBag.Hidden1 = "hidden = \"hidden\"";

}

<div id="carouselExampleIndicators" class="carousel slide mb-3" data-ride="carousel">

<ol class="carousel-indicators">

<li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>

<li data-target="#carouselExampleIndicators" data-slide-to="1"></li>

<li data-target="#carouselExampleIndicators" data-slide-to="2"></li>

</ol>

<div class="carousel-inner">

<div class="carousel-item active">

<img class="d-block w-100" src="~/Content/Images/home\_carusel1.jpg" alt="Первый слайд">

<div class="carousel-caption d-none d-md-block">

<div class="alert alert-warning text-dark" style="opacity: 0.7; filter: alpha(Opacity=70);" role="alert">

<h2>Делись тем, что любишь</h2>

<h4>Делитесь интересными историями, полезными советами и впечатляющими лайвхаками</h4>

</div>

</div>

</div>

<div class="carousel-item">

<img class="d-block w-100" src="~/Content/Images/home\_carusel2.jpg" alt="Второй слайд">

<div class="carousel-caption d-none d-md-block">

<div class="alert alert-warning text-dark" style="opacity: 0.7; filter: alpha(Opacity=70);" role="alert">

<h2>Занимайся творчеством</h2>

<h4>Выплесните все что у вас накипело в вашем блоге, возможно, у вас есть единомышленники ;)</h4>

</div>

</div>

</div>

<div class="carousel-item">

<img class="d-block w-100" src="~/Content/Images/home\_carusel3.jpg" alt="Третий слайд">

<div class="carousel-caption d-none d-md-block">

<div class="alert alert-warning text-dark" style="opacity: 0.7; filter: alpha(Opacity=70);" role="alert">

<h2>Придумывайте что-то новенькое</h2>

<h4>Старайтесь разнообразить свой контент, чтобы не потерять любимых читателей</h4>

</div>

</div>

</div>

</div>

<a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">

<span class="carousel-control-prev-icon" aria-hidden="true"></span>

<span class="sr-only">Назад</span>

</a>

<a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">

<span class="carousel-control-next-icon" aria-hidden="true"></span>

<span class="sr-only">Далее</span>

</a>

</div>

<h1 class="display-4">Хотите начать вести блог и стать популярным?</h1>

<p>

Тогда мы хотим предложить вашему вниманию наш удобный и простой сервис предназначенный специально для ведения и чтения блогов!

Здесь вы сможете быстро и просто создавать статьи, компоновать их по группам, чтобы

привлечь только заинтересованную публиику.

</p>

<p>

Чтобы начать свой блог прямо сейчас вам необходимо зарегистрировать свой аккаунт,

а затем перейти в раздел Блоги, чтобы создать статью на определенную тему.

</p>

<h1 class="display-4">Вы хотите просто почитать интересные статьи?</h1>

<p>

Вам тоже подходит MyBlog! Чтобы читать интересные статьи, забавные истории и полезные

лайфхаки пользователей со всей страны, вам даже не нужно регистрироваться на сайте.

Просто перейдите в Блоги, выберете интересующий вас раздел, а затем откройте понравившуюся статью.

</p>

<p>

Свои комментарии вы можете оставлять под статьей! Также вы можете ставить лайки, понравившимся статьям,

благодарить авторов и подписываться.

</p>

<blockquote class="blockquote text-right">

<p class="mb-0">Развивайтесь, делитесь, дружите и общайтесь в свое удовольствие!</p>

<footer class="blockquote-footer">С уважением, Администрация <cite title="MyBlog">MyBlog</cite></footer>

</blockquote>

Blogs/Index.cshtml

@using Lab1.Models;

@using System.Linq;

@using System.Data.Entity;

@model List<Article>

@{

ViewBag.Page2 = "nav-item active";

ViewBag.Hidden = "hidden = \"hidden\"";

}

@functions {

public string isActive(bool Condition)

{

if (Condition)

{

return "active";

}

else

{

return "";

}

}

public string isHidden(bool Condition)

{

if (Condition)

{

return "hidden = \"hidden\"";

}

else

{

return "";

}

}

public string colorMess(int? code)

{

if (code == 1)

{

return "danger";

}

return "primary";

}

public int getNextArtID(List<Article> articles, int ArticleID)

{

if (articles.Count() > 1)

{

for (int i = 0; i < articles.Count(); i++)

{

if (articles[i].ID == ArticleID)

{

if (i + 1 != articles.Count())

{

return articles[i + 1].ID;

}

if (i - 1 != -1)

{

return articles[i - 1].ID;

}

}

}

}

return 0;

}

}

@\*Вывод всех категорий в навигационной панели\*@

<ul class="nav nav-pills nav-fill mb-3" id="pills-tab" role="tablist">

<li class="nav-item">

<a class="nav-link @isActive(Model == null)" id="cat-id-0"

href="/Blogs/Index">Интересное</a>

</li>

@foreach (var cat in ViewBag.Categorys)

{

<li class="nav-item">

<a class="nav-link @isActive(cat.ID == ViewBag.categoryID)" id="cat-id-@cat.ID"

href="/Blogs/Index/?CategoryID=@cat.ID">@cat.Name</a>

</li>

}

</ul>

@\*Если передали категорию, то выводим содержимое вкладки "Интересное"\*@

@if (ViewBag.categoryID == 0)

{

//Оповещения

<div class="alert alert-@colorMess(ViewBag.MessageCode) col-12" @isHidden(ViewBag.Message == null) role="alert">

@ViewBag.Message

</div>

//Выводим несколько категорий с картинками

<div class="container">

<div class="row align-items-center">

<div class="col-4">

<a href="/Blogs/Index/?CategoryID=1">

<figure class="figure-img">

<img src="~/Content/Images/News.jpg" class="figure-img img-fluid rounded" alt="Новости">

<figcaption class="figure-caption text-center">Новости</figcaption>

</figure>

</a>

</div>

<div class="col-4">

<a href="/Blogs/Index/?CategoryID=2">

<figure class="figure-img">

<img src="~/Content/Images/IT.jpg" class="figure-img img-fluid rounded" alt="IT">

<figcaption class="figure-caption text-center">IT</figcaption>

</figure>

</a>

</div>

<div class="col-4">

<a href="/Blogs/Index/?CategoryID=3">

<figure class="figure-img">

<img src="~/Content/Images/HiTech.jpg" class="figure-img img-fluid rounded" alt="HiTech">

<figcaption class="figure-caption text-center">HiTech</figcaption>

</figure>

</a>

</div>

</div>

<div class="row align-items-center">

<div class="col-4">

<a href="/Blogs/Index/?CategoryID=6">

<figure class="figure-img">

<img src="~/Content/Images/Music.jpg" class="figure-img img-fluid rounded" alt="Музыка">

<figcaption class="figure-caption text-center">Музыка</figcaption>

</figure>

</a>

</div>

<div class="col-4">

<a href="/Blogs/Index/?CategoryID=5">

<figure class="figure-img">

<img src="~/Content/Images/Cinema.jpg" class="figure-img img-fluid rounded" alt="Кино">

<figcaption class="figure-caption text-center">Коно</figcaption>

</figure>

</a>

</div>

<div class="col-4">

<a href="/Blogs/Index/?CategoryID=7">

<figure class="figure-img">

<img src="~/Content/Images/Beauty.jpg" class="figure-img img-fluid rounded" alt="Красота и здоровье">

<figcaption class="figure-caption text-center">Красота и здоровье</figcaption>

</figure>

</a>

</div>

</div>

</div>

}

else //Если категория была указана в запросе, то открываем ее и вывоводим 10 последних статей

{

int categoryID = ViewBag.categoryID;

DbSet<User> authors = ViewBag.Authors;

DbSet<Category> categorys = ViewBag.Categorys;

DbSet<Favorites> favorites = ViewBag.Favorites;

Category one\_category = categorys.Where(x => x.ID == categoryID).ToList()[0];

<div class="container">

@\*Выводим заголовок категории и кнопку создания статьи\*@

<div class="row align-items-center">

<div class="col-md-9">

<h2 class="display-4 mb-3">@one\_category.Name</h2>

</div>

<div class="col-md-3 text-right mb-md-0 mb-2">

<a class="btn btn-primary" href="/Blogs/Create/?CategoryID=@categoryID">Создать статью</a>

</div>

</div>

@\*Выводим оповещение, если оно было\*@

<div class="alert alert-@colorMess(ViewBag.MessageCode) col-12" @isHidden(ViewBag.Message == null) role="alert">

@ViewBag.Message

</div>

@\*Выводим статьи\*@

@foreach (Article art in Model)

{

<div class="row" id="@("article" + art.ID)">

<div class="alert alert-secondary col-12" role="alert">

@\*Тема и набор кнопок\*@

<div class="row align-items-center">

<div class="col-lg-9 text-center">

<h3>@art.Subject</h3>

</div>

<div class="col-lg-3 text-center text-lg-right">

@\*Скрываем кнопки, если это не текущий пользователь\*@

<a class="btn btn-danger mb-1 btn-sm" href="/Blogs/Delete/?CategoryID=@art.CategoryID&ArticleID=@art.ID&page=@ViewBag.CurrentPage&NextArticleID=@getNextArtID(Model, art.ID)" @isHidden(art.AuthorID != 1)>Удалить</a>

<a class="btn btn-warning mb-1 btn-sm" href="/Blogs/Edit/?ArticleID=@art.ID&page=@ViewBag.CurrentPage&CategoryID=@art.CategoryID" @isHidden(art.AuthorID != 1)>Редактировать</a>

<button type="button" class="btn btn-secondary btn-sm mb-1" data-toggle="collapse"

href="#@("articleText" + art.ID)" aria-expanded="false"

aria-controls="@("articleText" + art.ID)">

Развернуть/Свернуть

</button>

</div>

</div>

<blockquote class="blockquote">

<hr />

@\*Сворачиваемое содержимое с текстом статьи\*@

<div class="collapse multi-collapse" id="@("articleText" + art.ID)">

<p>@Html.Raw(Html.Encode(art.Text).Replace("\r\n", "</p><p>"))</p>

<hr />

</div>

@\*Кнопка добавления в избранное, подпись автора, дата и время\*@

<div class="row align-items-center">

<div class="col-sm-4 text-right text-sm-left">

<a class="btn btn-info btn-sm" href="/Blogs/AddFavorites/?CurrentUserID=1&ArticleID=@art.ID&page=@ViewBag.CurrentPage"

@\*Если у текущего пользователя (в данном случае у меня с id=1) есть эта статья в избранном, то мы скрываем кнопку добавления данной статьи в избранное для этой статьи\*@

@isHidden(favorites.Where(x => x.UserID == 1 && x.ArticleID == art.ID).ToList().Count() >= 1)>Добавить в избранное</a>

</div>

<div class="col-sm-8">

@{

User author = authors.Where(x => x.ID == art.AuthorID).ToList()[0];

}

<footer class="blockquote-footer text-right">@author.LastName @author.FirstName - @art.DT.ToString("dd.MM.yyyy HH:mm") / <cite title="MyBlog">MyBlog</cite></footer>

</div>

</div>

</blockquote>

</div>

</div>

}

</div>

// Выводим преключатели между страницами, если статей больше 10

if (ViewBag.CountPage > 1)

{

<nav aria-label="Page navigation example">

<ul class="pagination justify-content-center">

@\*Ссылка перехода на предыдущую страницу\*@

<li class="page-item @ViewBag.FirstPageDiasbled">

<a class="page-link"

href="/Blogs/Index/?CategoryID=@ViewBag.categoryID&page=@(ViewBag.CurrentPage-1)">Назад</a>

</li>

@\*Если страниц не больше 5, то выводим ссылки на все\*@

@if (ViewBag.CountPage <= 5)

{

for (int i = 0; i < ViewBag.CountPage; i++)

{

<li class="page-item @isActive(ViewBag.CurrentPage==i+1)"><a class="page-link" href="/Blogs/Index/?CategoryID=@categoryID&page=@(i+1)">@(i + 1)</a></li>

}

}

else

{

// Если страниц больше 5, то выводим ссылки только 5 ближайших

// Если текущая страница имеет слева и справа хотя бы по 2, то

// Выводим слева и справа от текущей по 2 страницы

if (ViewBag.CurrentPage > 2 && ViewBag.CurrentPage < ViewBag.CountPage - 1)

{

for (int i = ViewBag.CurrentPage - 3; i < ViewBag.CurrentPage + 2; i++)

{

<li class="page-item @isActive(ViewBag.CurrentPage == i + 1)"><a class="page-link" href="/Blogs/Index/?CategoryID=@categoryID&page=@(i + 1)">@(i + 1)</a></li>

}

}

else

{

// Если страниц меню слева от текущей 1 или меньше,

// то выводим первые 5 ссылок на страницы

if (ViewBag.CurrentPage <= 2)

{

for (int i = 0; i < 5; i++)

{

<li class="page-item @isActive(ViewBag.CurrentPage == i + 1)"><a class="page-link" href="/Blogs/Index/?CategoryID=@categoryID&page=@(i + 1)">@(i + 1)</a></li>

}

}

else

{

// Если страниц меню справа от текущей 1 или меньше,

// то выводим последние 5 ссылок на страницы

if (ViewBag.CurrentPage >= ViewBag.CountPage - 1)

{

for (int i = ViewBag.CountPage - 5; i < ViewBag.CountPage; i++)

{

<li class="page-item @isActive(ViewBag.CurrentPage==i+1)"><a class="page-link" href="/Blogs/Index/?CategoryID=@categoryID&page=@(i+1)">@(i + 1)</a></li>

}

}

}

}

}

@\*Ссылка перехода на следующую страницу\*@

<li class="page-item @ViewBag.LastPageDiasbled">

<a class="page-link"

href="/Blogs/Index/?CategoryID=@ViewBag.categoryID&page=@(ViewBag.CurrentPage+1)">Далее</a>

</li>

</ul>

</nav>

}

}

### Модели

Также были созданы модели

Article

namespace Lab1.Models

{

public class Article

{

public int ID { get; set; }

public int CategoryID { get; set; }

public int AuthorID { get; set; }

public DateTime DT { get; set; }

public string Subject { get; set; }

public string Text { get; set; }

}

}

Author

namespace Lab1.Models

{

public class Author

{

public int ID { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string MiddleName { get; set; }

public DateTime BirthDay { get; set; }

public string Email { get; set; }

public string PassHash { get; set; }

public string Phone { get; set; }

public bool Man { get; set; }

}

}

Category

namespace Lab1.Models

{

public class Category

{

public int ID { get; set; }

public string Name { get; set; }

}

}

## Примеры работы программы

На рисунке 1 представлена Главная страница сайта.

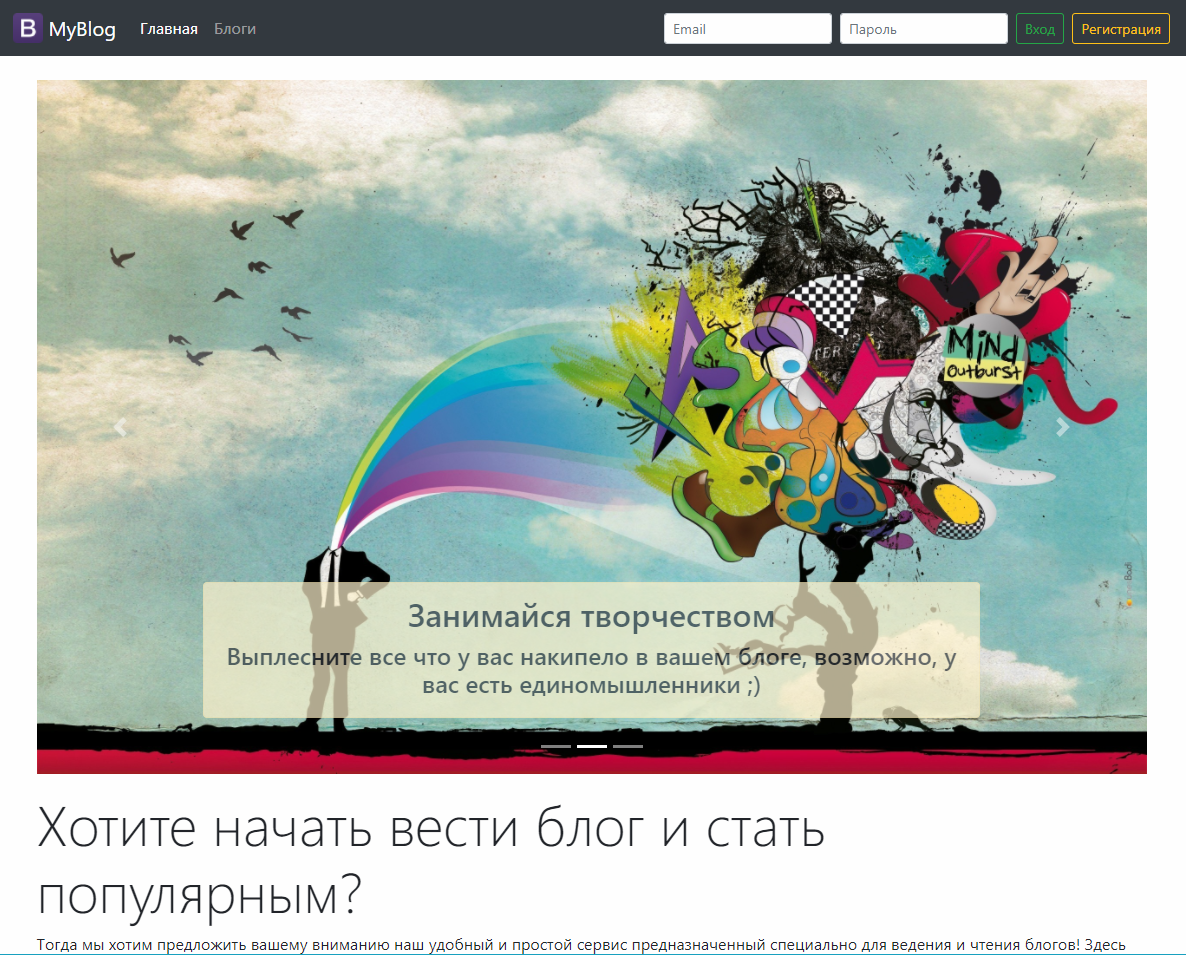


Рисунок 1 – Главная страница

На рисунке 2 представлена страница Блогов в категории Новости.

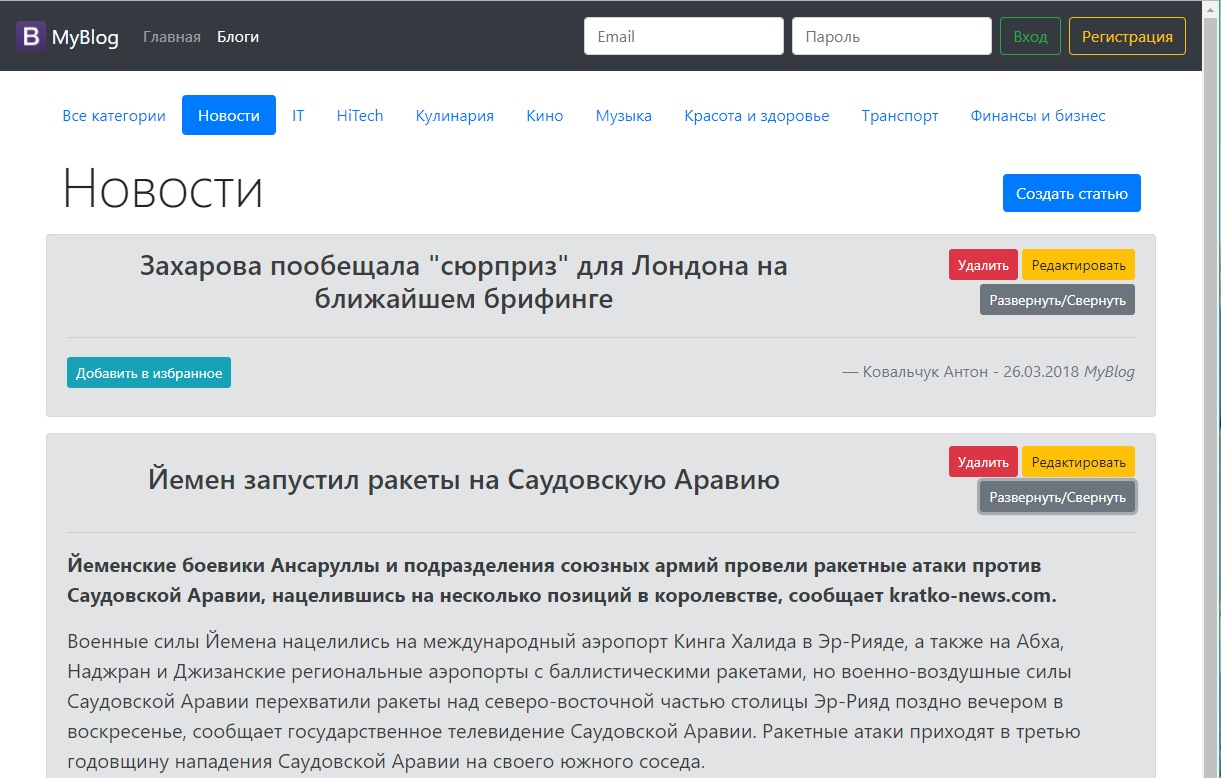


Рисунок 2 – Страница Блогов в разделе Новости

На рисунке 3 представлена страница Регистрации пользователя.

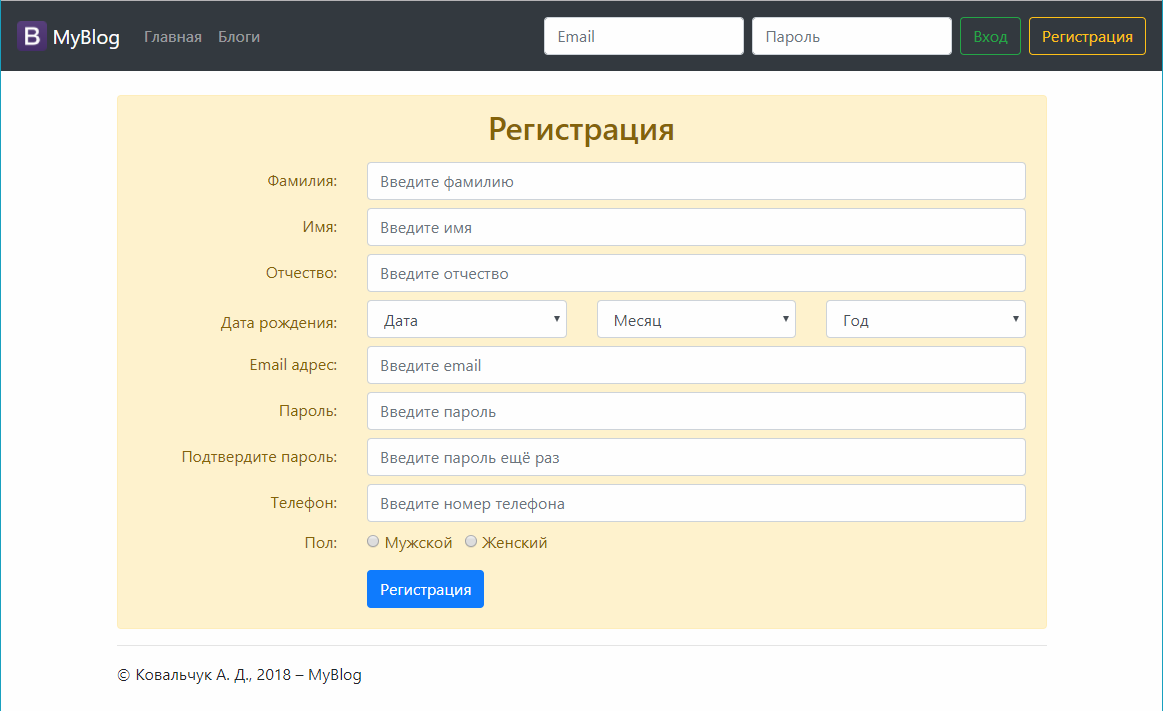


Рисунок 3 – Страница Регистрации пользователя

На рисунке 4 представлена форма для редактирования статьи

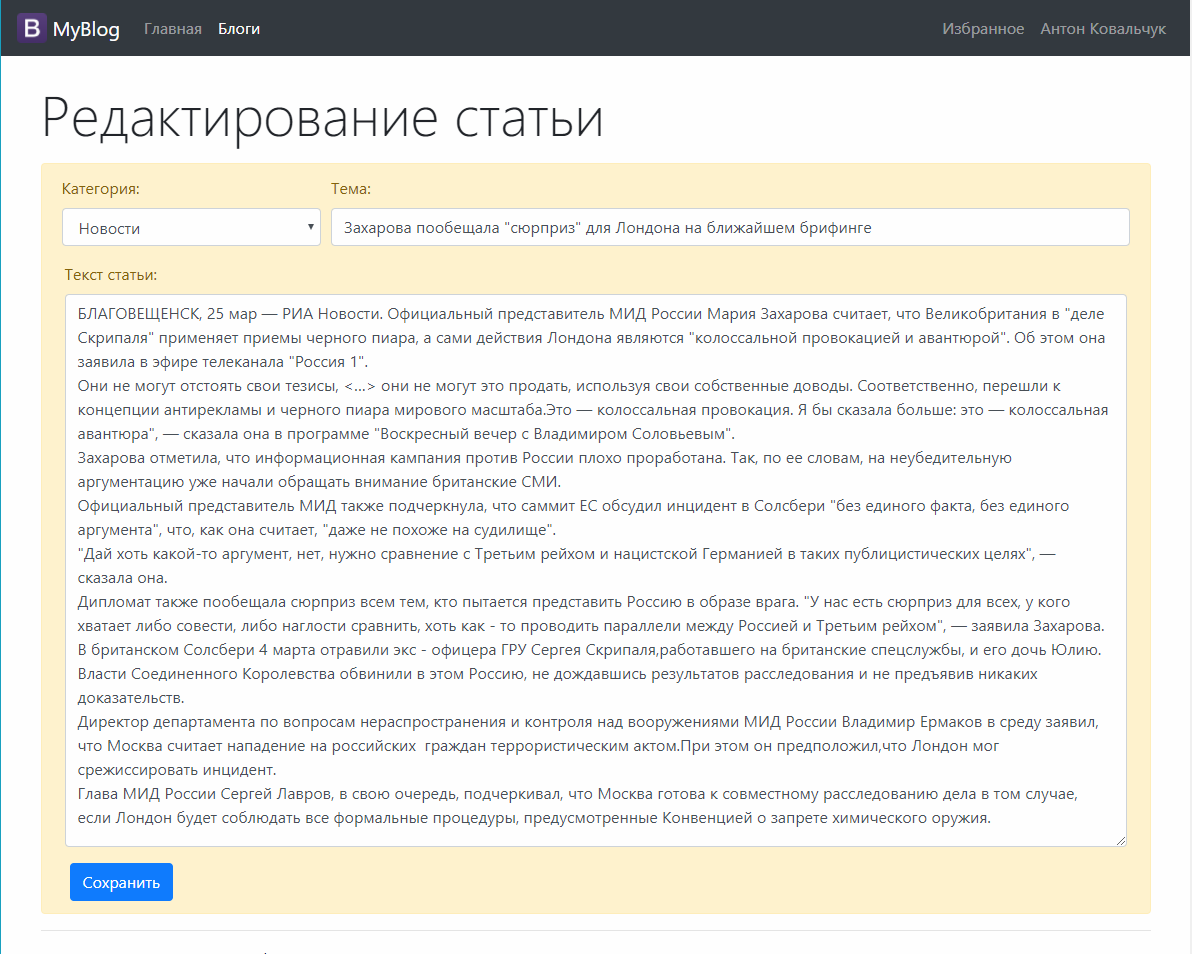


Рисунок 4 – Форма для редактирования

На рисунке 5 представлена форма для добавления статьи

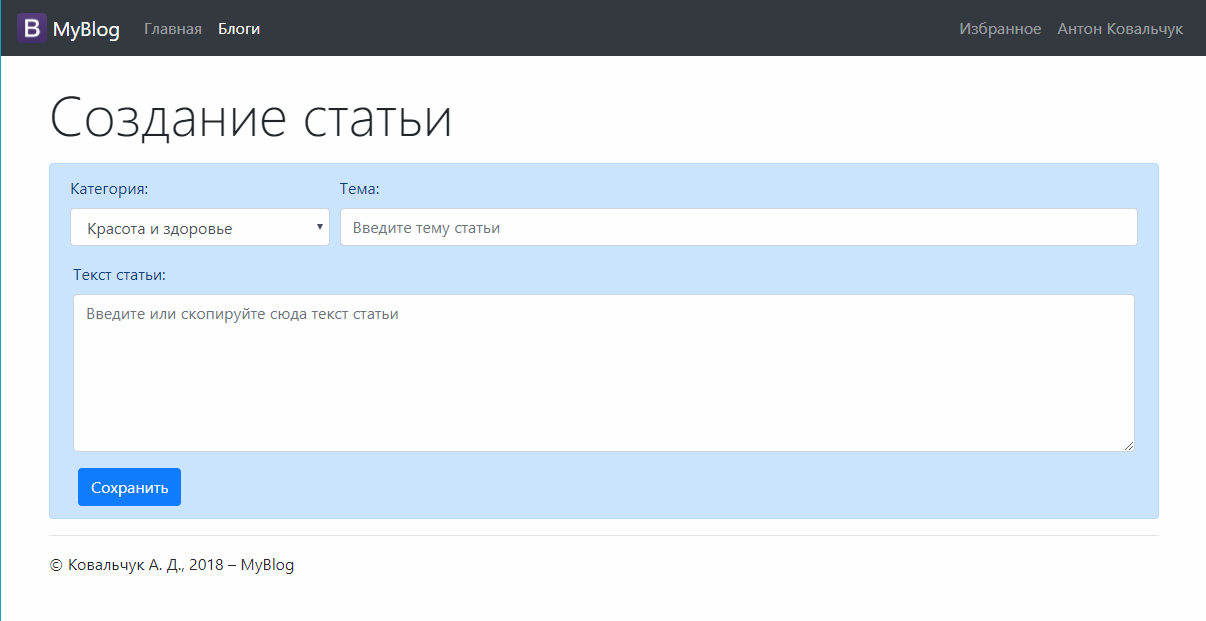


Рисунок 5 – Форма для добавления статьи

На рисунке 6 представлена страница «Избранное»

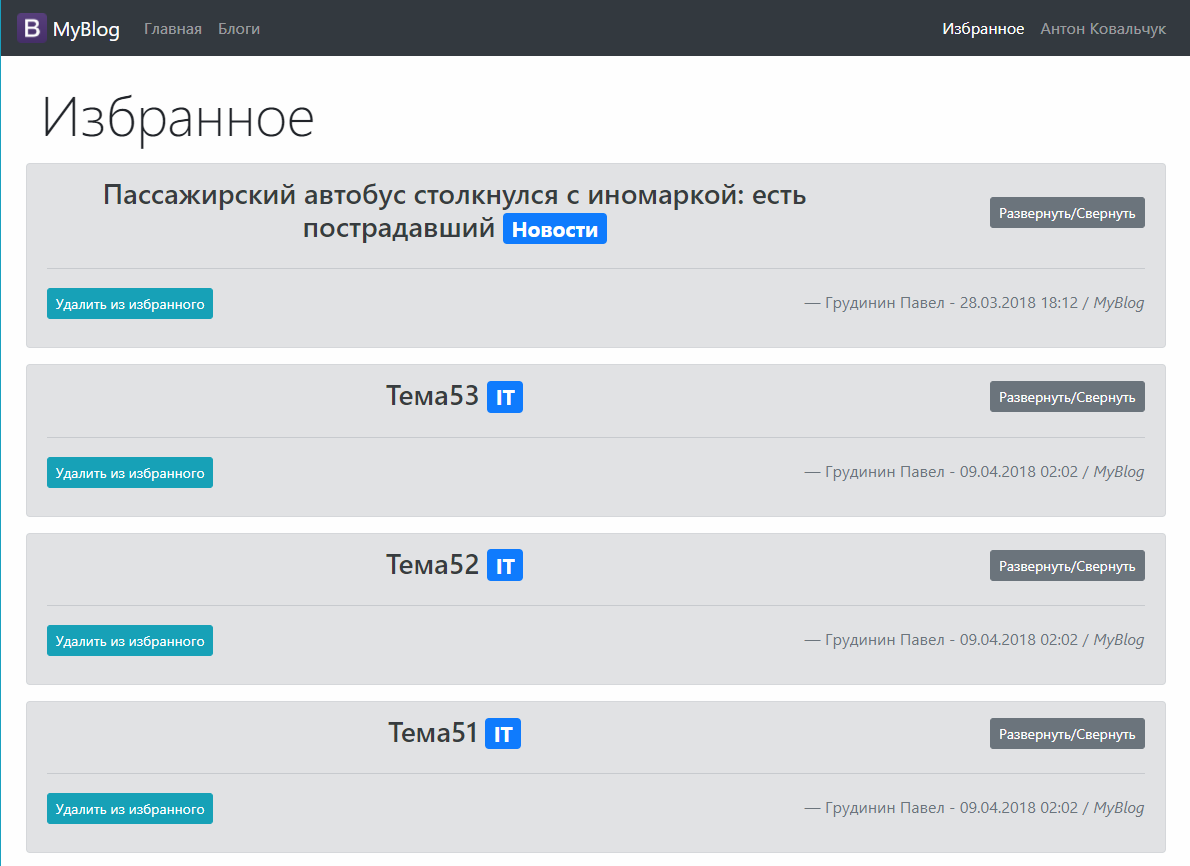


Рисунок 6 – Страница «Избранное»