|  |  |  |  |
| --- | --- | --- | --- |
| Модули | Описание | Количество строк кода | Размер (в Кбайтах) |
| 1 | 2 | 3 | 4 |
| BasketController.cs |  | 19 | 1 |
| CorpusController.cs |  | 113 | 4 |
| DDRController.cs |  | 93 | 3 |
| HomeController.cs |  | 42 | 1 |
| LoginController.cs |  | 70 | 3 |
| MemoryController.cs |  | 94 | 3 |
| MotherboardController.cs |  | 94 | 3 |
| OhladController.cs |  | 93 | 3 |
| PcBuilderController.cs |  | 142 | 5 |
| PowerBlockController.cs |  | 93 | 3 |
| ProcessorController.cs |  | 93 | 3 |
| VideocardController.cs |  | 93 | 3 |

1. BasketController.cs

﻿using Config.Repository;

using Config.Repository.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Config.Controllers

{

public class BasketController : Controller

{

// GET: Basket

public ActionResult Index()

{

return View(new KorzinaView());

}

}

}

2. CorpusController.cs

﻿using Config.Models;

using Config.Repository;

using Config.Repository.Models;

using Microsoft.Ajax.Utilities;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Identity.Client;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Threading.Tasks;

using System.Web;

using System.Web.Mvc;

using System.Web.Services.Description;

namespace Config.Controllers

{

public class CorpusController : Controller

{

private PCBuilderDbContext db = new PCBuilderDbContext();

[HttpGet]

public ActionResult Create()

{

return View();

}

public ActionResult Get()

{

return View(db.Corpus);

}

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Repository.Models.Corpus corpus = db.Corpus.Find(id);

if (corpus == null)

{

return HttpNotFound();

}

return View(corpus);

}

[HttpPost]

public ActionResult Delete(int id)

{

Repository.Models.Corpus corpus = db.Corpus.Find(id);

db.Corpus.Remove(corpus);

db.SaveChanges();

return RedirectToAction("Get", "Corpus");

}

public ActionResult Edit(int Id)

{

var corp = db.Corpus.Where(s => s.IdCorpus == Id).FirstOrDefault();

return View(corp);

}

[HttpPost]

public ActionResult Edit(Repository.Models.Corpus corpus, HttpPostedFileBase ImageFile)

{

var corp = db.Corpus.Where(s => s.IdCorpus == corpus.IdCorpus).FirstOrDefault();

byte[] imageData = null;

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

corpus.Image = imageData;

db.Corpus.Remove(corp);

db.Corpus.Add(corpus);

db.SaveChanges();

return RedirectToAction("Get","Corpus");

}

[HttpPost]

public ActionResult Create(Repository.Models.Corpus corpus, HttpPostedFileBase ImageFile)

{

//if (ModelState.IsValid)

//{

// if (img != null)

// {

// corpus.Image = new byte[img.ContentLength];

// img.InputStream.Read(corpus.Image, 0, img.ContentLength);

// }

// db.Corpus.Add(corpus);

// db.SaveChanges();

//}

byte[] imageData = null;

// считываем переданный файл в массив байтов

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

// установка массива байтов

corpus.Image = imageData;

db.Corpus.Add(corpus);

db.SaveChanges();

ModelState.Clear();

return View();

}

}

}

3. DDRController.cs

﻿using Config.Repository;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

namespace Config.Controllers

{

public class DDRController : Controller

{

private PCBuilderDbContext db = new PCBuilderDbContext();

public ActionResult Create()

{

return View();

}

public ActionResult Get()

{

return View(db.Ddr);

}

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Repository.Models.Ddr ddr = db.Ddr.Find(id);

if (ddr == null)

{

return HttpNotFound();

}

return View(ddr);

}

[HttpPost]

public ActionResult Delete(int id)

{

Repository.Models.Ddr ddr = db.Ddr.Find(id);

db.Ddr.Remove(ddr);

db.SaveChanges();

return RedirectToAction("Get", "DDR");

}

public ActionResult Edit(int Id)

{

var ddr = db.Ddr.Where(s => s.IdDdr == Id).FirstOrDefault();

return View(ddr);

}

[HttpPost]

public ActionResult Edit(Repository.Models.Ddr ddr, HttpPostedFileBase ImageFile)

{

var dr = db.Ddr.Where(s => s.IdDdr == ddr.IdDdr).FirstOrDefault();

byte[] imageData = null;

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

ddr.Image = imageData;

db.Ddr.Remove(dr);

db.Ddr.Add(ddr);

db.SaveChanges();

return RedirectToAction("Get", "DDR");

}

[HttpPost]

public ActionResult Create(Repository.Models.Ddr ddr, HttpPostedFileBase ImageFile)

{

byte[] imageData = null;

// считываем переданный файл в массив байтов

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

// установка массива байтов

ddr.Image = imageData;

db.Ddr.Add(ddr);

db.SaveChanges();

ModelState.Clear();

return RedirectToAction("Get","DDR");

}

}

}

4. HomeController.cs

﻿using Config.Models;

using Config.Repository.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Config.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

return View();

}

[Authorize (Roles = "1,2")]

public ActionResult AdminPanel()

{

return View();

}

public ActionResult PcBuilder()

{

return View();

}

public ActionResult About()

{

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

ViewBag.Message = "Your contact page.";

return View();

}

}

}

5. LoginController.cs

﻿using Config.Repository;

using Config.Repository.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Security;

namespace Config.Controllers

{

public class LoginController : Controller

{

private PCBuilderDbContext db = new PCBuilderDbContext();

// GET: Login

public ActionResult Register()

{

return View();

}

//POST: Login

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Register(User user)

{

if (ModelState.IsValid)

{

db.User.Add(user);

db.SaveChanges();

ModelState.Clear();

}

return View();

}

public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(Login login)

{

using (PCBuilderDbContext db = new PCBuilderDbContext()) ;

{

var user = db.User.Where(a => a.Login == login.Log && a.Password == login.Password).FirstOrDefault();

if (user != null)

{

var Ticket = new FormsAuthenticationTicket(login.Log, true, 3000);

string Encrypt = FormsAuthentication.Encrypt(Ticket);

var cookie = new HttpCookie(FormsAuthentication.FormsCookieName, Encrypt);

cookie.Expires= DateTime.Now.AddHours(3000);

cookie.HttpOnly = true;

Response.Cookies.Add(cookie);

if(user.RoleId == 1)

{

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

}

else

{

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

}

}

}

return View();

}

public ActionResult Logout()

{

FormsAuthentication.SignOut();

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

}

}

}

6. MemoryController.cs

﻿using Config.Repository;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

namespace Config.Controllers

{

public class MemoryController : Controller

{

private PCBuilderDbContext db = new PCBuilderDbContext();

public ActionResult Create()

{

return View();

}

public ActionResult Get()

{

return View(db.Memory);

}

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Repository.Models.Memory memory = db.Memory.Find(id);

if (memory == null)

{

return HttpNotFound();

}

return View(memory);

}

[HttpPost]

public ActionResult Delete(int id)

{

Repository.Models.Memory memory = db.Memory.Find(id);

db.Memory.Remove(memory);

db.SaveChanges();

return RedirectToAction("Get", "Memory");

}

public ActionResult Edit(int Id)

{

var memory = db.Memory.Where(s => s.IdMemory == Id).FirstOrDefault();

return View(memory);

}

[HttpPost]

public ActionResult Edit(Repository.Models.Memory memory, HttpPostedFileBase ImageFile)

{

var mem = db.Memory.Where(s => s.IdMemory == memory.IdMemory).FirstOrDefault();

byte[] imageData = null;

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

memory.Image = imageData;

db.Memory.Remove(mem);

db.Memory.Add(memory);

db.SaveChanges();

return RedirectToAction("Get", "Memory");

}

[HttpPost]

public ActionResult Create(Repository.Models.Memory memory, HttpPostedFileBase ImageFile)

{

byte[] imageData = null;

// считываем переданный файл в массив байтов

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

// установка массива байтов

memory.Image = imageData;

db.Memory.Add(memory);

db.SaveChanges();

ModelState.Clear();

return RedirectToAction("Get", "Memory");

}

}

}

7. MotherboardController.cs

﻿using Config.Repository;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

namespace Config.Controllers

{

public class MotherboardController : Controller

{

private PCBuilderDbContext db = new PCBuilderDbContext();

public ActionResult Create()

{

return View();

}

public ActionResult Get()

{

return View(db.Motherboard);

}

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Repository.Models.Motherboard motherboard = db.Motherboard.Find(id);

if (motherboard == null)

{

return HttpNotFound();

}

return View(motherboard);

}

[HttpPost]

public ActionResult Delete(int id)

{

Repository.Models.Motherboard motherboard = db.Motherboard.Find(id);

db.Motherboard.Remove(motherboard);

db.SaveChanges();

return RedirectToAction("Get", "Motherboard");

}

public ActionResult Edit(int Id)

{

var memory = db.Motherboard.Where(s => s.IdMotherboard == Id).FirstOrDefault();

return View(memory);

}

[HttpPost]

public ActionResult Edit(Repository.Models.Motherboard motherboard, HttpPostedFileBase ImageFile)

{

var mother = db.Motherboard.Where(s => s.IdMotherboard == motherboard.IdMotherboard).FirstOrDefault();

byte[] imageData = null;

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

motherboard.Image = imageData;

db.Motherboard.Remove(mother);

db.Motherboard.Add(motherboard);

db.SaveChanges();

return RedirectToAction("Get", "Motherboard");

}

[HttpPost]

public ActionResult Create(Repository.Models.Motherboard motherboard, HttpPostedFileBase ImageFile)

{

byte[] imageData = null;

// считываем переданный файл в массив байтов

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

// установка массива байтов

motherboard.Image = imageData;

db.Motherboard.Add(motherboard);

db.SaveChanges();

ModelState.Clear();

return RedirectToAction("Get", "Motherboard");

}

}

}

8. OhladController.cs

﻿using Config.Repository;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

namespace Config.Controllers

{

public class OhladController : Controller

{

private PCBuilderDbContext db = new PCBuilderDbContext();

public ActionResult Create()

{

return View();

}

public ActionResult Get()

{

return View(db.Ohlad);

}

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Repository.Models.Ohlad ohlad = db.Ohlad.Find(id);

if (ohlad == null)

{

return HttpNotFound();

}

return View(ohlad);

}

[HttpPost]

public ActionResult Delete(int id)

{

Repository.Models.Ohlad ohlad = db.Ohlad.Find(id);

db.Ohlad.Remove(ohlad);

db.SaveChanges();

return RedirectToAction("Get", "Ohlad");

}

public ActionResult Edit(int Id)

{

var ohlad = db.Ohlad.Where(s => s.IdOhlad == Id).FirstOrDefault();

return View(ohlad);

}

[HttpPost]

public ActionResult Edit(Repository.Models.Ohlad ohlad, HttpPostedFileBase ImageFile)

{

var oh = db.Ohlad.Where(s => s.IdOhlad == ohlad.IdOhlad).FirstOrDefault();

byte[] imageData = null;

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

ohlad.Image = imageData;

db.Ohlad.Remove(oh);

db.Ohlad.Add(ohlad);

db.SaveChanges();

return RedirectToAction("Get", "Ohlad");

}

[HttpPost]

public ActionResult Create(Repository.Models.Ohlad ohlad, HttpPostedFileBase ImageFile)

{

byte[] imageData = null;

// считываем переданный файл в массив байтов

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

// установка массива байтов

ohlad.Image = imageData;

db.Ohlad.Add(ohlad);

db.SaveChanges();

ModelState.Clear();

return RedirectToAction("Get", "Ohlad");

}

}

}

9. PcBuilderController.cs

﻿using Config.Repository;

using Config.Repository.Models;

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Config.Controllers

{

public class PcBuilderController : Controller

{

private PCBuilderDbContext db = new PCBuilderDbContext();

// GET: PcBuilder

public ActionResult Index()

{

var corpus = db.Corpus.ToList();

var processors = db.Processor.ToList();

var ddrs = db.Ddr.ToList();

var memories = db.Memory.ToList();

var motherboards = db.Motherboard.ToList();

var ohlads = db.Ohlad.ToList();

var powerBlocks = db.PowerBlock.ToList();

var videocards = db.Videocard.ToList();

var model = new Builder { Corpus = corpus, Processors = processors, Ddrs = ddrs, Memorys = memories, Motherboards = motherboards, Ohlads = ohlads, PowerBlocks = powerBlocks, Videocards = videocards};

return View(model);

}

[HttpPost]

public ActionResult AddToBuildVideokarta()

{

int id = int.Parse(Request.Form["id"]);

var VK = db.Videocard.SingleOrDefault(s => s.IdVideocard == id);

if(VK!=null)

{

Korzina.videocardKorzina = VK;

}

return Redirect("~/Basket");

}

[HttpPost]

public ActionResult AddToBuildPowerblock()

{

int id = int.Parse(Request.Form["id"]);

var VK = db.PowerBlock.SingleOrDefault(s => s.IdPowerblock == id);

if (VK != null)

{

Korzina.powerBlockKorzina = VK;

}

return Redirect("~/Basket");

}

[HttpPost]

public ActionResult AddToBuildProcessor()

{

int id = int.Parse(Request.Form["id"]);

var VK = db.Processor.SingleOrDefault(s => s.IdProcessor == id);

if (VK != null)

{

Korzina.processorKorzina = VK;

}

return Redirect("~/Basket");

}

[HttpPost]

public ActionResult AddToBuildMotherboard()

{

int id = int.Parse(Request.Form["id"]);

var VK = db.Motherboard.SingleOrDefault(s => s.IdMotherboard == id);

if (VK != null)

{

Korzina.motherboardKorzina = VK;

}

return Redirect("~/Basket");

}

[HttpPost]

public ActionResult AddToBuildDdr()

{

int id = int.Parse(Request.Form["id"]);

var VK = db.Ddr.SingleOrDefault(s => s.IdDdr == id);

if (VK != null)

{

Korzina.ddrKorzina = VK;

}

return Redirect("~/Basket");

}

[HttpPost]

public ActionResult AddToBuildMemory()

{

int id = int.Parse(Request.Form["id"]);

var VK = db.Memory.SingleOrDefault(s => s.IdMemory == id);

if (VK != null)

{

Korzina.memoryKorzina = VK;

}

return Redirect("~/Basket");

}

[HttpPost]

public ActionResult AddToBuildOhlad()

{

int id = int.Parse(Request.Form["id"]);

var VK = db.Ohlad.SingleOrDefault(s => s.IdOhlad == id);

if (VK != null)

{

Korzina.ohladKorzina = VK;

}

return Redirect("~/Basket");

}

[HttpPost]

public ActionResult AddToBuildCorpus()

{

int id = int.Parse(Request.Form["id"]);

var VK = db.Corpus.SingleOrDefault(s => s.IdCorpus == id);

if (VK != null)

{

Korzina.corpusKorzina = VK;

}

return Redirect("~/Basket");

}

}

}

10. PowerBlockController.cs

﻿using Config.Repository;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

namespace Config.Controllers

{

public class PowerBlockController : Controller

{

private PCBuilderDbContext db = new PCBuilderDbContext();

public ActionResult Create()

{

return View();

}

public ActionResult Get()

{

return View(db.PowerBlock);

}

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Repository.Models.PowerBlock powerblock = db.PowerBlock.Find(id);

if (powerblock == null)

{

return HttpNotFound();

}

return View(powerblock);

}

[HttpPost]

public ActionResult Delete(int id)

{

Repository.Models.PowerBlock powerblock = db.PowerBlock.Find(id);

db.PowerBlock.Remove(powerblock);

db.SaveChanges();

return RedirectToAction("Get", "PowerBlock");

}

public ActionResult Edit(int Id)

{

var power = db.PowerBlock.Where(s => s.IdPowerblock == Id).FirstOrDefault();

return View(power);

}

[HttpPost]

public ActionResult Edit(Repository.Models.PowerBlock powerblock, HttpPostedFileBase ImageFile)

{

var pow = db.PowerBlock.Where(s => s.IdPowerblock == powerblock.IdPowerblock).FirstOrDefault();

byte[] imageData = null;

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

powerblock.Image = imageData;

db.PowerBlock.Remove(pow);

db.PowerBlock.Add(powerblock);

db.SaveChanges();

return RedirectToAction("Get", "PowerBlock");

}

[HttpPost]

public ActionResult Create(Repository.Models.PowerBlock powerblock, HttpPostedFileBase ImageFile)

{

byte[] imageData = null;

// считываем переданный файл в массив байтов

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

// установка массива байтов

powerblock.Image = imageData;

db.PowerBlock.Add(powerblock);

db.SaveChanges();

ModelState.Clear();

return RedirectToAction("Get", "PowerBlock");

}

}

}

11. ProcessorController.cs

﻿using Config.Repository;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

namespace Config.Controllers

{

public class ProcessorController : Controller

{

private PCBuilderDbContext db = new PCBuilderDbContext();

public ActionResult Create()

{

return View();

}

public ActionResult Get()

{

return View(db.Processor);

}

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Repository.Models.Processor processor = db.Processor.Find(id);

if (processor == null)

{

return HttpNotFound();

}

return View(processor);

}

[HttpPost]

public ActionResult Delete(int id)

{

Repository.Models.Processor processor = db.Processor.Find(id);

db.Processor.Remove(processor);

db.SaveChanges();

return RedirectToAction("Get", "Processor");

}

public ActionResult Edit(int Id)

{

var processor = db.Processor.Where(s => s.IdProcessor == Id).FirstOrDefault();

return View(processor);

}

[HttpPost]

public ActionResult Edit(Repository.Models.Processor processor, HttpPostedFileBase ImageFile)

{

var proc = db.Processor.Where(s => s.IdProcessor == processor.IdProcessor).FirstOrDefault();

byte[] imageData = null;

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

processor.Image = imageData;

db.Processor.Remove(proc);

db.Processor.Add(processor);

db.SaveChanges();

return RedirectToAction("Get", "Processor");

}

[HttpPost]

public ActionResult Create(Repository.Models.Processor processor, HttpPostedFileBase ImageFile)

{

byte[] imageData = null;

// считываем переданный файл в массив байтов

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

// установка массива байтов

processor.Image = imageData;

db.Processor.Add(processor);

db.SaveChanges();

ModelState.Clear();

return RedirectToAction("Get", "Processor");

}

}

}

12. VideocardController.cs

﻿using Config.Repository;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Web;

using System.Web.Mvc;

namespace Config.Controllers

{

public class VideocardController : Controller

{

private PCBuilderDbContext db = new PCBuilderDbContext();

public ActionResult Create()

{

return View();

}

public ActionResult Get()

{

return View(db.Videocard);

}

public ActionResult Delete(int? id)

{

if (id == null)

{

return new HttpStatusCodeResult(HttpStatusCode.BadRequest);

}

Repository.Models.Videocard videocard = db.Videocard.Find(id);

if (videocard == null)

{

return HttpNotFound();

}

return View(videocard);

}

[HttpPost]

public ActionResult Delete(int id)

{

Repository.Models.Videocard videocard = db.Videocard.Find(id);

db.Videocard.Remove(videocard);

db.SaveChanges();

return RedirectToAction("Get", "Videocard");

}

public ActionResult Edit(int Id)

{

var videocard = db.Videocard.Where(s => s.IdVideocard == Id).FirstOrDefault();

return View(videocard);

}

[HttpPost]

public ActionResult Edit(Repository.Models.Videocard videocard, HttpPostedFileBase ImageFile)

{

var proc = db.Videocard.Where(s => s.IdVideocard == videocard.IdVideocard).FirstOrDefault();

byte[] imageData = null;

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

videocard.Image = imageData;

db.Videocard.Remove(proc);

db.Videocard.Add(videocard);

db.SaveChanges();

return RedirectToAction("Get", "Videocard");

}

[HttpPost]

public ActionResult Create(Repository.Models.Videocard videocard, HttpPostedFileBase ImageFile)

{

byte[] imageData = null;

// считываем переданный файл в массив байтов

using (var binaryReader = new BinaryReader(ImageFile.InputStream))

{

imageData = binaryReader.ReadBytes(ImageFile.ContentLength);

}

// установка массива байтов

videocard.Image = imageData;

db.Videocard.Add(videocard);

db.SaveChanges();

ModelState.Clear();

return RedirectToAction("Get", "Videocard");

}

}

}