*Online Shopping Store*

Izradio: *Hrvoje Cimaš*

Sadržaj

[1. Opis projektnog zadatka 3](#_Toc530156080)

[2. Baza 4](#_Toc530156081)

[3. Dijagrami razreda 5](#_Toc530156082)

[4. Plan rada 11](#_Toc530156083)

[5. Korisničko sučelje 12](#_Toc530156084)

[Prijava administratora 13](#_Toc530156085)

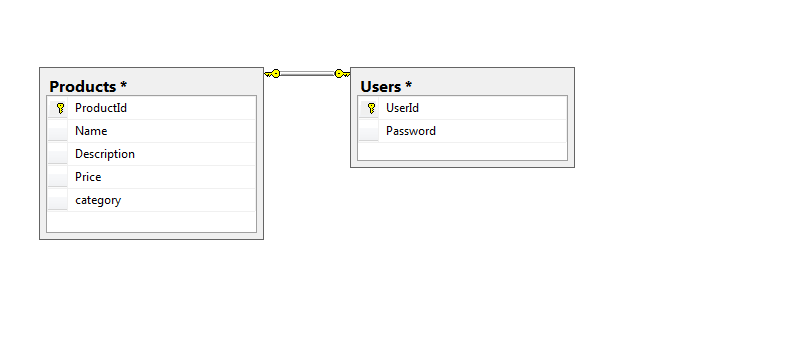
[6. Zaključci i smjernice za daljnji rad 15](#_Toc530156086)

# 1. Opis projektnog zadatka

Cilj ovog projektnog zadatka je bio napraviti Web aplikaciju. U ovom slučaju izrađena je stranica pod nazivom "Online Shopping Store" koja se bavi prodajom elektroničke opreme putem interneta. Proizvodi se mogu sortirati prema odgovarajućoj kategoriji. Naglasak web stranice je na jednostavnom dizajnu. Aplikacija ima mogućnost pomoću koje korisnici mogu kupiti proizvod, te pomoću administracijske prijave mogućnost izmjene postojećih i dodavanja novih proizvoda. Ključne točke kod implementacije aplikacije su autorizacija korisnika, izrada narudžba, pretraživanje narudžba, te ispis podataka . Korisnici koji bi mogli biti zainteresirani za ovu aplikaciju su dućani koji nude narudžbu i dostavu putem interneta.

# 2. Baza

Ovdje možemo vidjeti koji podaci se nalaze u bazi



# 3. Dijagrami razreda

*Klasa Product u kojoj se nalaze svi podaci vezani za svaki proizvod*

public class Product

{

**public int ProductId { get; set; }**

**[Required(ErrorMessage = "Name is required.")]**

**public string Name { get; set; }**

**[Required(ErrorMessage = "Description is required.")]**

**public string Description { get; set; }**

**public decimal Price{get; set;}**

**[Required(ErrorMessage="Category is required.")]**

**public string Category { get; set; }**

}

ProductController omogućava prikaz proizvoda po stranicama

public class ProductController : Controller

{

private readonly SkladisteProizvoda repository;

public int PageSize = 4;

public ProductController(SkladisteProizvoda repo)

{

repository = repo;

}

public ViewResult List(string category, int page = 1)

{

ProductsListViewModel model = new ProductsListViewModel

{

Products = repository.Products.Where(p => category == null || p.Category == category).OrderBy(p => p.ProductId).Skip((page - 1) \* PageSize).Take(PageSize),

PagingInfo = new PagingInfo

{

CurrentPage = page,

ItemsPerPage = PageSize,

TotalItems = category == null ? repository.Products.Count() :

repository.Products.Where(p => p.Category == category).Count()

},

CurrentCategory = category

};

return View(model);

}

}

Klasa Cart i CartController nam omogućuju dodavanje i brisanje proizvoda iz naše košarice, pregled proizvoda te slanje naše narudžbe.

public class Cart

{

private List<CartLine> lineCollection = new List<CartLine>();

public void AddItem(Product product, int quantity)

{

CartLine line = lineCollection.Where(p => p.Product.ProductId == product.ProductId).FirstOrDefault();

if (line == null)

{

lineCollection.Add(new CartLine { Product = product, Quantity = quantity });

}

else

{

line.Quantity += quantity;

}

}

public void RemoveLine(Product product)

{

lineCollection.RemoveAll(p => p.Product.ProductId == product.ProductId);

}

public decimal ComputeTotalValue()

{

return lineCollection.Sum(p => p.Product.Price \* p.Quantity);

}

public IEnumerable<CartLine> Lines

{

get { return lineCollection; }

}

public void Clear()

{

lineCollection.Clear();

}

}

CartController

public class CartController : Controller

{

private SkladisteProizvoda repository;

public CartController(SkladisteProizvoda repo)

{

repository = repo;

}

public ViewResult Index(Cart cart, string returnUrl)

{

return View(new CartIndexViewModel { Cart = cart, ReturnUrl = returnUrl });

}

public PartialViewResult Summary(Cart cart)

{

return PartialView(cart);

}

public ViewResult Checkout()

{

ViewBag.Message = "Your order has been sent successfully";

return View(new ShippingDetails());

}

public RedirectToRouteResult AddToCart(Cart cart, int productId, string returnUrl)

{

Product product = repository.Products.FirstOrDefault(p => p.ProductId == productId);

if(product != null)

{

cart.AddItem(product, 1);

}

return RedirectToAction("Index", new { returnUrl});

}

public RedirectToRouteResult RemoveFromCart(Cart cart, int productId, string returnUrl)

{

Product product = repository.Products.FirstOrDefault(p => p.ProductId == productId);

if(product != null)

{

cart.RemoveLine(product);

}

return RedirectToAction("Index", new {returnUrl});

}

}

AdminController omogućava dodavanje, izmjenu i brisanje proizvoda

public class AdminController : Controller

{

private SkladisteProizvoda repository;

public AdminController(SkladisteProizvoda repo)

{

repository = repo;

}

public ActionResult Index()

{

return View(repository.Products);

}

public ViewResult Create()

{

return View(new Product());

}

[HttpPost]

public ActionResult Create(Product product)

{

if (ModelState.IsValid)

{

repository.SavaProduct(product);

TempData["message"] = string.Format("{0} has been saved", product.Name);

return RedirectToAction("Index");

}

else

{

return View(product);

}

}

public ViewResult Edit(int productId)

{

Product product = repository.Products.FirstOrDefault(p => p.ProductId == productId);

return View(product);

}

[HttpPost]

public ActionResult Edit(Product product)

{

if (ModelState.IsValid)

{

repository.SavaProduct(product);

TempData["message"] = string.Format("{0} has been saved", product.Name);

return RedirectToAction("Index");

}

else

{

return View(product);

}

}

[HttpPost]

public ActionResult Delete(int productId)

{

Product deletedProduct = repository.DeleteProduct(productId);

if(deletedProduct != null)

{

TempData["message"] = string.Format("{0} was deleted", deletedProduct.Name);

}

return RedirectToAction("Index");

} }

AccountControler služi za prijavu i odjavu administratora

public class AccountController : Controller

{

Autentifikacija autentifikacija;

public AccountController(Autentifikacija autentifikacija)

{

this.autentifikacija = autentifikacija;

}

[AllowAnonymous]

public ActionResult Login()

{

return View();

}

[HttpPost]

[AllowAnonymous]

public ActionResult Login(LoginViewModel model, string ReturnUrl)

{

if (ModelState.IsValid)

{

if(autentifikacija.Autentifikacija(model.UserName, model.Password ) )

{

FormsAuthentication.SetAuthCookie(model.UserName, false);

return Redirect(ReturnUrl ?? Url.Action("Index", "Admin"));

}

else

{

ModelState.AddModelError("", "Incorrect username or password");

return View();

}

}

else

{

return View();

}

}

public ActionResult Logout()

{

FormsAuthentication.SignOut();

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

}

}

}

# 4. Plan rada

Plan rada:

Do 28.07.2017. odraditi dio projekta vezan za prikaz proizvoda.

Do 05.08.2017. odraditi dio projekta vezan za slanje proizvoda.

Do 15.08.2017. odraditi dio projekta vezan za pretraživanje proizvoda po kategorijama

Do 20.08.2017. odraditi dio projekta vezan za logiranje, mijenjanje postojećih proizvoda i dodavanje novih.

Tokom izrade projekta za prikaz početne strane i prikaz svih proizvoda nije bilo problema, sve je obavljeno u roku.

Tokom izrade dijela projekta vezanog za slanje proizvoda, bilo je manjih poteškoća pa je taj dio obavljen uz malo zakašnjenje.

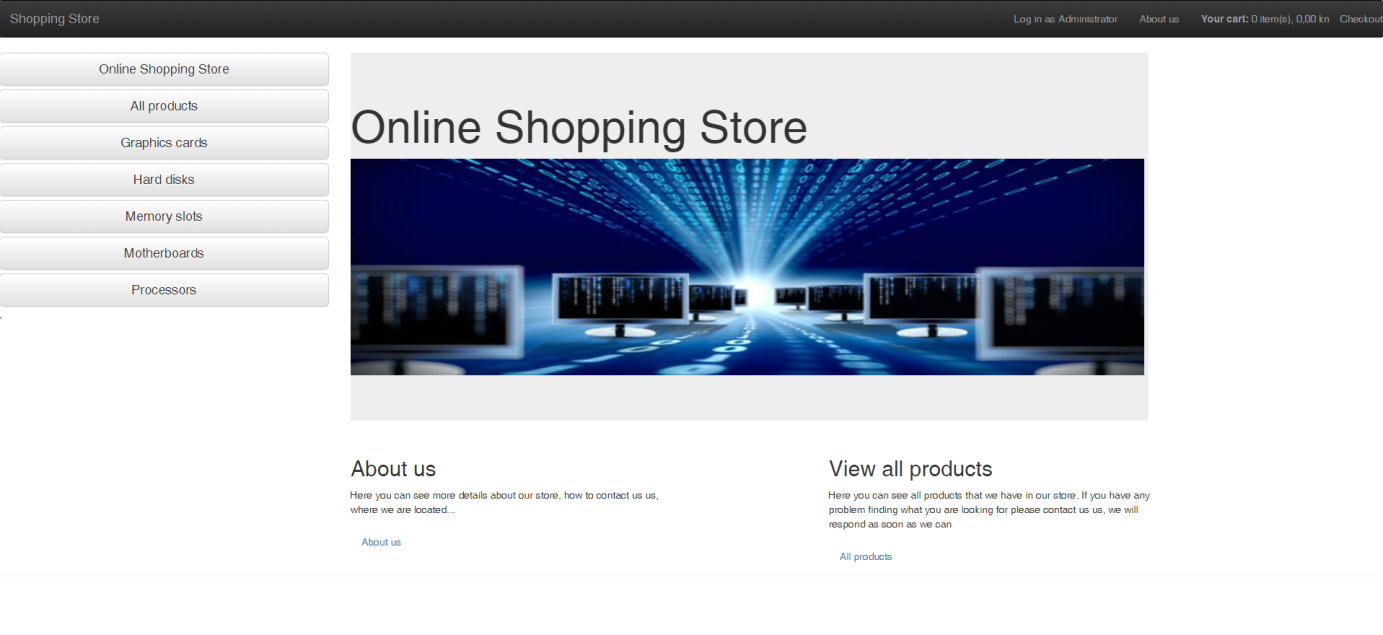
Dio vezan za pretraživanje proizvoda po kategorijama je uspješno riješen na vrijeme

Sa dijelom projekta vezanog uz logiranje nije bilo problema, ali za opcije koje su vezane za naknadno mijenja proizvoda i dodavanje novih je potrajalo duže nego što je za to bilo predviđeno.

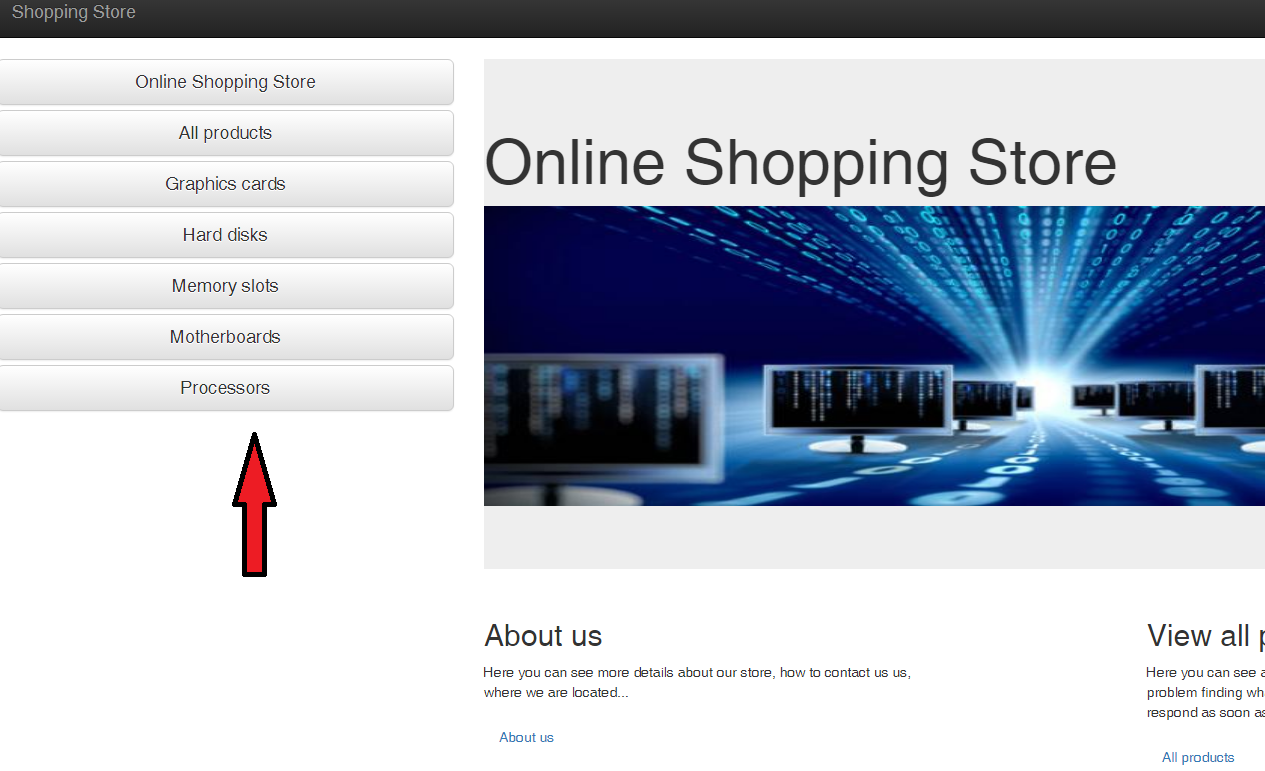
# 5. Korisničko sučelje

U ovom poglavlju se nalaze prikazi izgleda zaslona

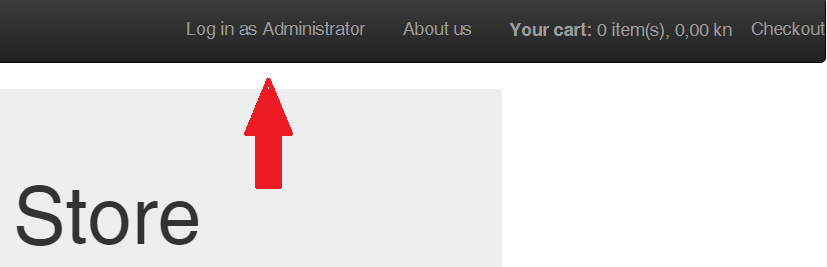
Početna strana



Preko izbornika se može brzo i jednostavno odabrati željena kategorija za one proizvode koji nas zanimaju



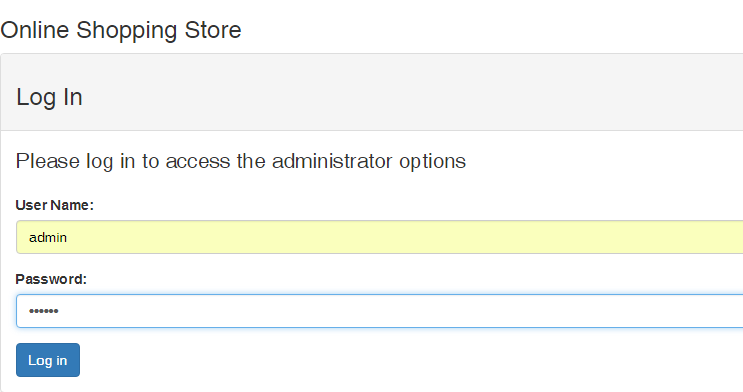
Login in as administrator - omogućava prijavu administratora, kako bi on mogao dodavati nove i brisati postojeće proizvode

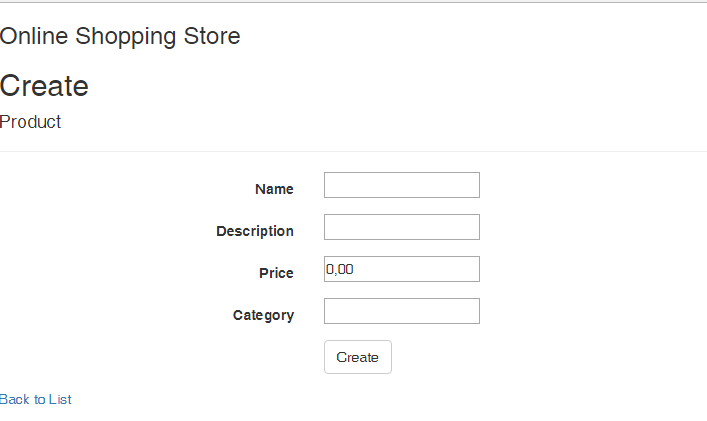


## Prijava administratora

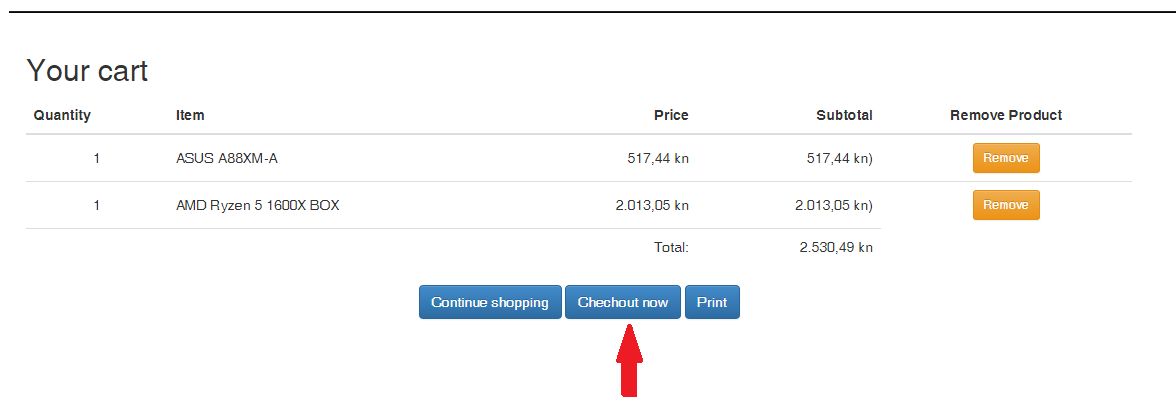
User Name: admin

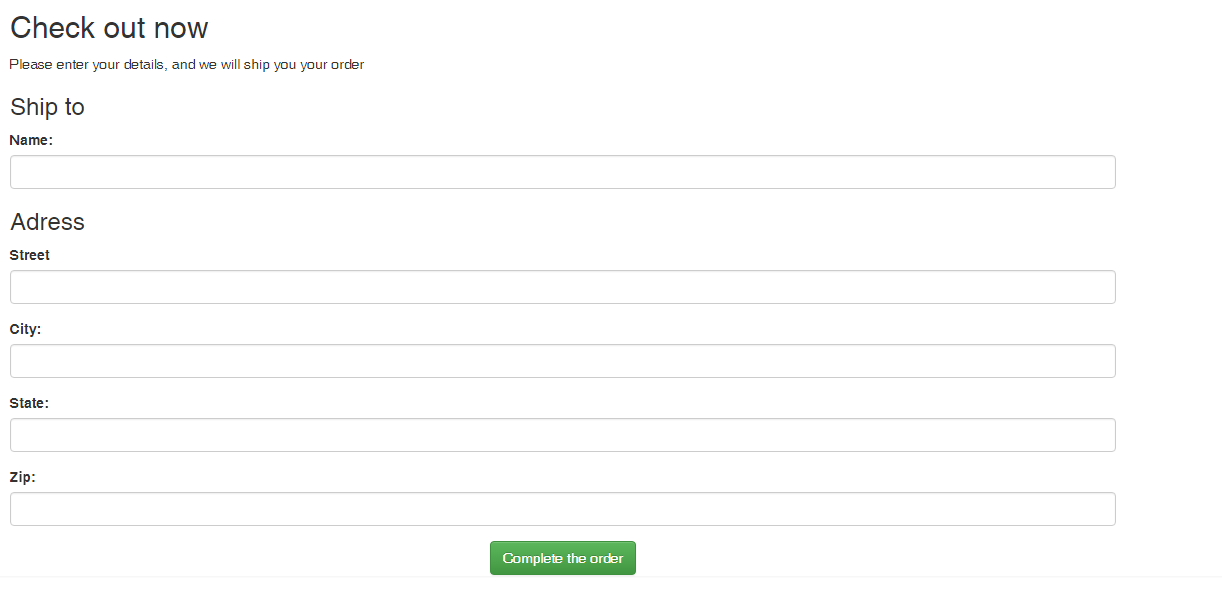
Password: 123456



Dodavanje novog proizvoda

Košarica sa našim proizvodima - klikom na gumb Continue shopping" nastavljamo sa našom kupovinom , klikom na gumb "Print" možemo isprintati našu stranicu sa proizvodima, gumb "Checkout" vodi nas na obrazac kojeg ispunjujemo sa našim podacima





# 6. Zaključci i smjernice za daljnji rad

Tokom razvoja projekta najviše vremena je potrošeno na sortiranje proizvoda te na mogućnosti administratora koje on može obavljati (dodavanje novih, izmjena i brisanje postojećih proizvoda). Kao nadogradnja projekta mogla bi se uvesti mogućnost pomoću koje bi korisnici mogli objavljivati svoja iskustva sa ovom Web trgovinom, komentirati jesu li zadovoljni te kako bi se mogla poboljšati.