From Wipro_.Net_Azure_SD_APRC1.

Great Learning Platform.

Capstone Project

Design and Implementation of E-Commerce sites for Toys Shopping

Toy Story!

By

Group – 3

Anant Agrawal

Malika Dutta

Maheswari Thanniru

Tulika Srivastava

Rashi Agarwal

List of Contents:

Abstract 3

1.0 Introduction

- **1.1. Examples of Ecommerce Website and business** 4
- **1.2. How E-commerce Websites Work** 4
- **1.3. Economic Benefits** 6
- **1.4. Who can start an ecommerce store** 6
- **1.5. Job Scope** 7

2.0 Overall description

- **2.1 Description** 7
- **2.2 Using the code** 7
- **2.3 Web Pages Details** 8
- **2.4 Project Details** 8

3.0 System Requirements

- **3.1 Use-Case Diagram** 9

4.0 Design & Implementation

- **4.1. Home Page** 10
- **4.2. Login Page** 10
  - **a. Admin Login** 11
  - **b. User Login** 11
- **4.3. User Register Page** 12
- **4.4. Admin Home Page** 12
- **4.5. Products Page** 13
- **4.6. My cart** 14
- **4.7. Billing & Address Page** 14
- **4.8. Transaction gateway Page** 15

5.0 Azure DevOps 17

6.0 Code Snippets 18

7.0 Conclusion 28

Abstract:

Recently the e-commerce platform is playing an important role in some areas; its activities are a subset of e-business activities. The aim is to build and develop a reliable website based on the e-commerce theories, developing effective well designed web pages. This website is about selling different toys. In order to implement the online website, we need to use current technologies to achieve this goal. We need to set up an online e-commerce store which is easy-to-use. Further, it's important to improve the customer experience. Lastly, implement the direct Online Sale between businesses to consumers by implementing electronic payment methods. All these techniques should be based on a deliberate plan according to the strategy of e-commerce while implementing the current technologies to ensure a good revenue to the company.

In today's fast-changing business environment, it's extremely important to be able to respond to client needs in the most effective and timely manner. If your customers wish to see your business online and have instant access to your products or services. Online Shopping is a lifestyle e-commerce web application, which retails various fashion and lifestyle products. This project allows viewing various products available and enables registered users to purchase desired products by placing orders using the instant payment option or Cash on Delivery (Pay Later) option. This project provides easy access to Administrators and Managers to view orders placed. In order to develop an e-commerce website, a number of Technologies must be studied and understood. These include multi-tiered architecture, server and client side scripting techniques, implementation technologies such as ASP.NET, programming language (such as C#) and relational databases. This is a project with the objective to develop a basic website where a consumer is provided with a shopping cart application and also to know about the technologies used to develop such an application.

1.0 Introduction:

An e-commerce website is an information technology method in which traders, businesses/distributors/marketers can sell products/services and the customer can purchase on that website electronically by using the internet on the mobile and computer. It means an e-commerce website is an online shop where 'e' means electronic and commerce means business. The website is a group of HTML web pages that is created to market/sell information/product/services.

From a bigger perspective, every website on the internet is the e - Commerce Website. It can be the platform, it can be a marketplace, it can be a portal, it can be apps, it can be an entertainment website, shopping website, online courses website and online degree college.

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

The objective of this project is to develop a general purpose e-commerce toy store where products can be bought from the comfort of home through the Internet.

An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address and payment information.

1.1 Examples of Ecommerce Website and business:

- When you purchase a mobile phone/ shoes/ software/ flowers on any website such as Amazon, Flip-kart, etc. and pay through credit/debit card or cash on delivery and then the seller delivers the product through courier or post mail on your location then it's called e-commerce. In this case, Flip-kart is an online store website or an e-commerce website.
- When you subscribe to watch a cricket match, movies, and shows on any website such as Hotstar through debit card and credit card it is called e-commerce. And in this way, Hotstar is a digital and mobile entertainment e-commerce website.
- When you rent or buy movies on YouTube and pay to watch them by using the mobile/ computer and internet it's called e-commerce. In this method, you have used your computer/ mobile and internet through electricity and visited youtube website to watch/buy/rent the movie and paid through debit card/credit card/ net banking/ payment wallet etc. It means youtube is an eCommerce website in which you can buy/ watch/ rent the latest movies and shows.
- When you use Google Ads or Facebook Advertising etc. to promote and advertise your products/services online and pay Google and Facebook to use the services and platforms then it's e-commerce. In this case, Google / Facebook etc. are e-commerce

companies that provide you platform and tools to advertise and promote your business/products/services online.

- When you launch your apps on the Google Play store, you pay Google to use their platform to connect with your customers/target audience then it's e-commerce and google's play store is an eCommerce platform in which apps developers or apps launcher or owners have to pay Google. And all these transactions are online. It's e-commerce.
- When you recharge your mobile phone/dish TV/ internet data pack by using the internet and the website such as Paytm, mobiwiki, JIO etc. and pay through debit/credit card, wallet, and net banking then it's e-commerce and PayTM, MobiKwik, Jio apps or websites are e-commerce websites. In which they are doing customers recharge online and get paid directly to their bank account.
- When you purchase software as a service, platform as a service, infrastructure as service for your business from cloud computing service providers such as Alibaba, Amazon web service, Microsoft, google cloud etc. on their website then these are the e-commerce websites.
- When you use internet banking then it is e-commerce. You pay bills, transfer money, open an RD/FD account, pay installments online, pay for offline products from payment wallets etc. are e-commerce.
- When political parties, government or non-government organizations received funds or donations online then it is e-commerce.

Examples of an e-commerce website:

While every website is an e-commerce website and platform. But the following are well known or those are considered e-commerce websites.

- Amazon
- Flipkart
- Shopclues
- ajio.com Etc.

Every website is an e-commerce website and platform on the Internet. And even the rise of the internet is only because of e-commerce. It's impossible to imagine the internet without e-commerce. E-Commerce is the main source of wealth for internet companies and online businesses. If there is no commerce then there is no internet. Ecommerce is now a part of our daily life.

1.2 How E-commerce Websites Works:

Almost all E-Commerce websites work similarly. Following is the process of e-commerce website/e**c**ommerce business/online transactions.

Internet – Connecting the people through computers/mobiles and the internet.

User – Searching on Google and other search engines for products/services and daily life solutions.

Website – After search or research user visits the website that is in the top 10 search results.

Products / Services – User (customer) finds the product and selects it and adds it to the cart.

Purchased – Now users purchase it through debit and credit cards by using a third-party payment gateway such as ccavenue, payubiz etc.

Payment Gateway – (Payment Gateway, Merchant Accounts, and Online Credit Card Processing Service provider) – Received the payment and transfer it into website owner or sellers accounts after 1 week or later.

Bank Account – Customer can pay using the debit/credit card and net banking/merchant receive money in the bank account from payment gateway service provider.

Delivery – on spot, While many deliver the product within 8 days after receiving the payment. Some only receive cash on the delivery.

1.3 Economic benefits:

- Banks are earning money due to the increased use of debit/credit cards.
- The government can use that money for the development of its own machinery or citizens' development.

Business benefits:

- Less costly to sell products and services.
- A wider variety of customers on the internet than offline stores/shops.
- No credits.
- Easy to manage transactions.
- Easy to market/sell.

Consumer Benefits:

- Getting product/service at the door.
- Saves time.
- Less costly than offline.
- To showcase modernism.

1.4 Who can start an ecommerce store:

Anyone can start or build an ecommerce website such as shopkeepers, handmade items manufacturers, small and medium domestic product manufacturers etc.

- **Business Scope:** Business scope is unlimited. More than half of the population on the earth is on the Internet intentionally and unintentionally. People are looking for the best quality and organic products, they want to feel vow at less cost without wasting time on offline shopping. So if someone thinks that they can sell online or have anything that people need then they can start.

1.5 Job scope:

E-Commerce created and increased the Information Technology related jobs scopes worldwide. The purpose of an e-commerce website is to sell more and earn a profit. More and

more people trying to build and run an online store and due to that there is a demand for the following experts:

- Ecommerce or online store builder or ecommerce website designer
- Content writer to write product descriptions and website content.
- Search Engine Optimizer to rank the website higher on the google.
- Graphics Designer to create high-quality images.
- Digital Marketer who handles or manages search engine marketing, social media marketing and video marketing etc.
- Data analyst who can help ecommerce store owners to make a decision based on customer behavior and data analytics such as what kind of products they can sell more, which page on the ecommerce website needs to be updated, what customers are looking for and which landing page is working better than others.

2.0 Overall Description:

2.1 Description:

- Any member can register and view available products.
- Only registered members can purchase multiple products.
- Customer Care page is available to contact Admin for queries.
- There are three roles available: Visitor, User and Admin.
  • Visitors can view available products.
  • Users can view and purchase products.
  • An Admin has some extra privilege including all privileges of visitor and user.
    - Admin can add products, edit product information and add/remove products.
    - Admin can edit user information and can remove users.
    - Admin can ship orders to users based on orders placed.

2.2 Using the code:

1. Attach the database in your "SQL Server Management Studio Express".

2. Run the application on Microsoft Visual Studio as a web site.

3. Locate the database.

Technology used:

Front end: MVC Core MVC Application.

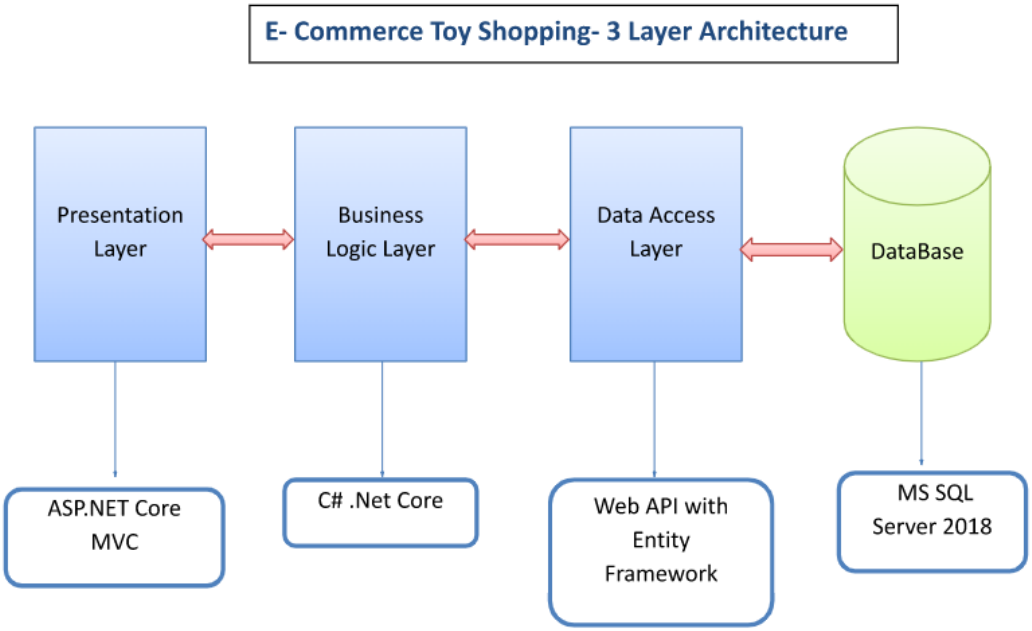
Middleware: Asp.NET Core Web API with Entity Framework Database first approach

Backend: Database: MS Sql Server

2.3 Web Pages details:

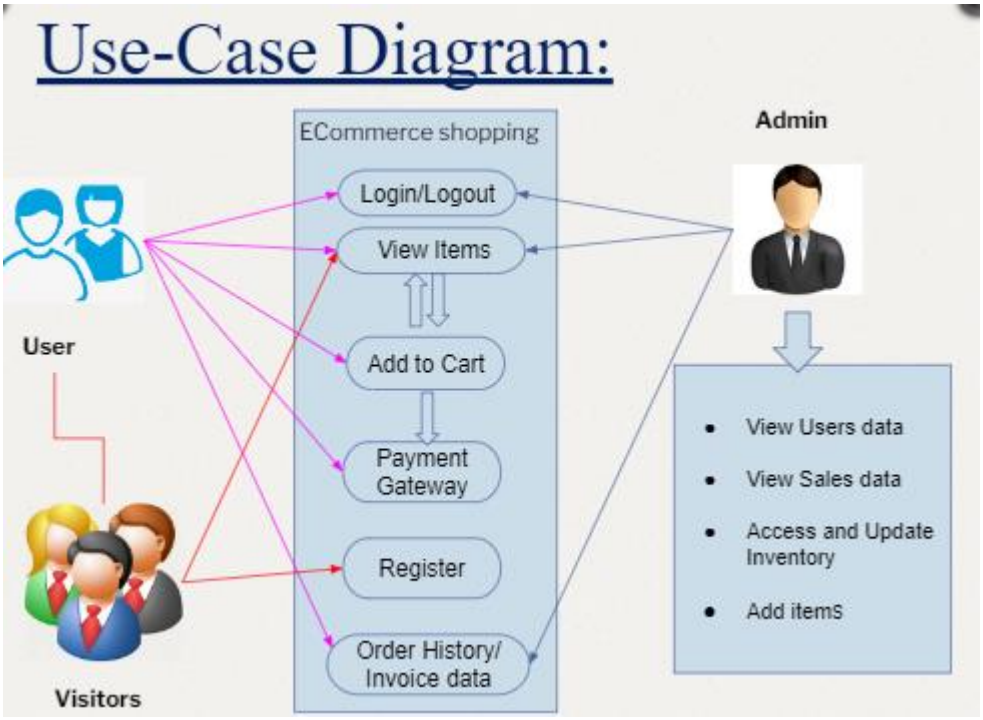
- Home Page.
- Login Page.
  - -Admin Page
  - -User Page
- New User Register Page.
- Admin Home Page.
  - -User data view
  - -Sales/Transaction Data view
  - -Add New Product
  - -Inventory view
- Category of Products Page.
  - - Product data View
- My Cart
- Billing & Address Page.
- Transaction Gateway Page.
- Payment Confirmation Page
- Logout View

2.4 Project Details:



3.0 System Requirements:

3.1Use-Case Diagram:

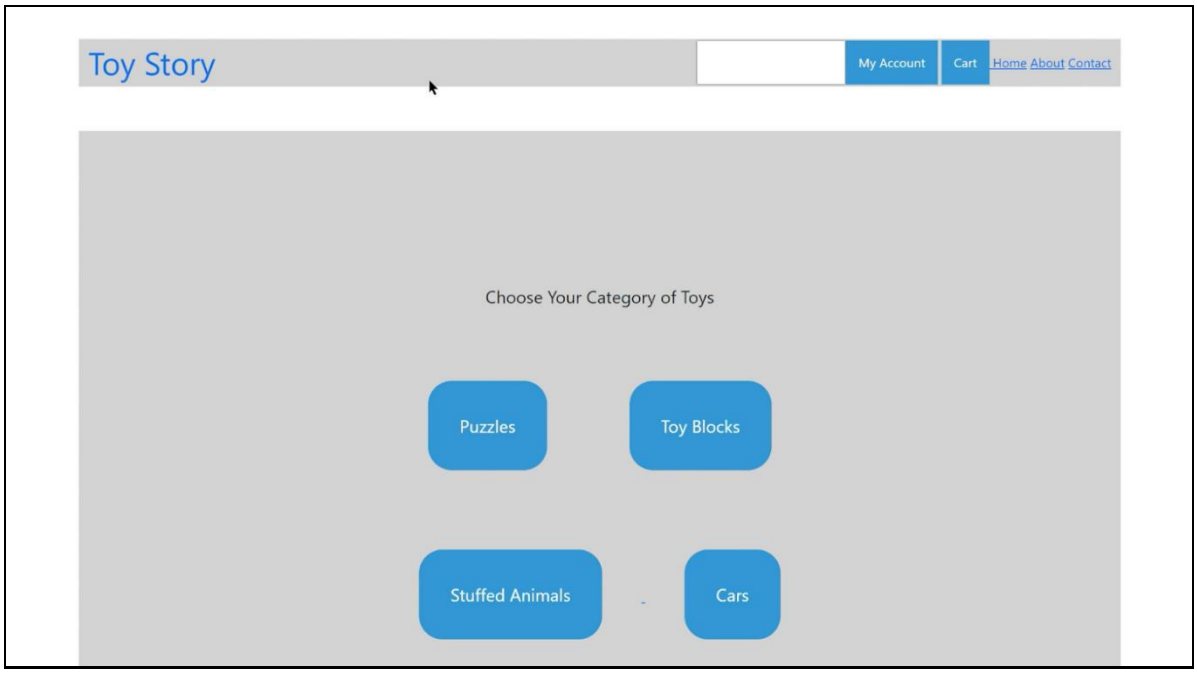


4.0 Design & Implementation:

Anyone can view an ecommerce toy shopping portal and available products, but every user must login by his/her Username and password in order to purchase or order products. Unregistered members can register by navigating to registration page. Only Admin will have access to modify items, by default developer can only be an 'Admin'. Once a user registers a site, his default role will be 'User'.

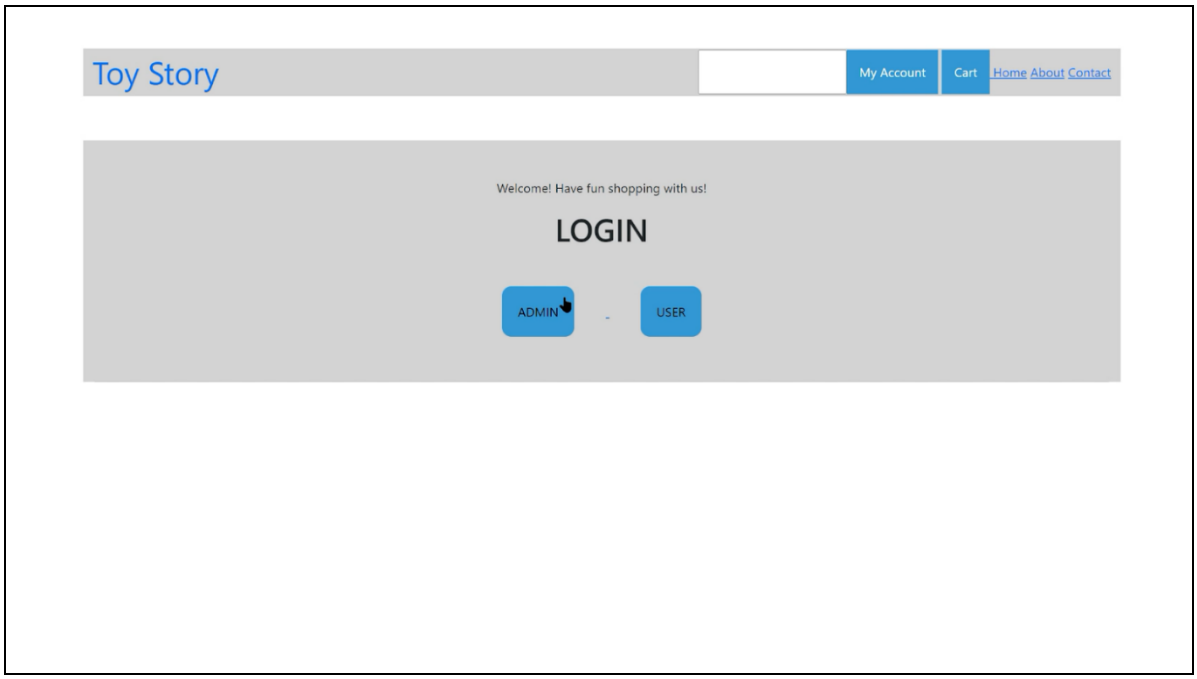
4.1 Home Page:

The Home Screen will consist of a screen where one can browse through the categories of toys which we have on our website. Press on any category and view the respective products page.



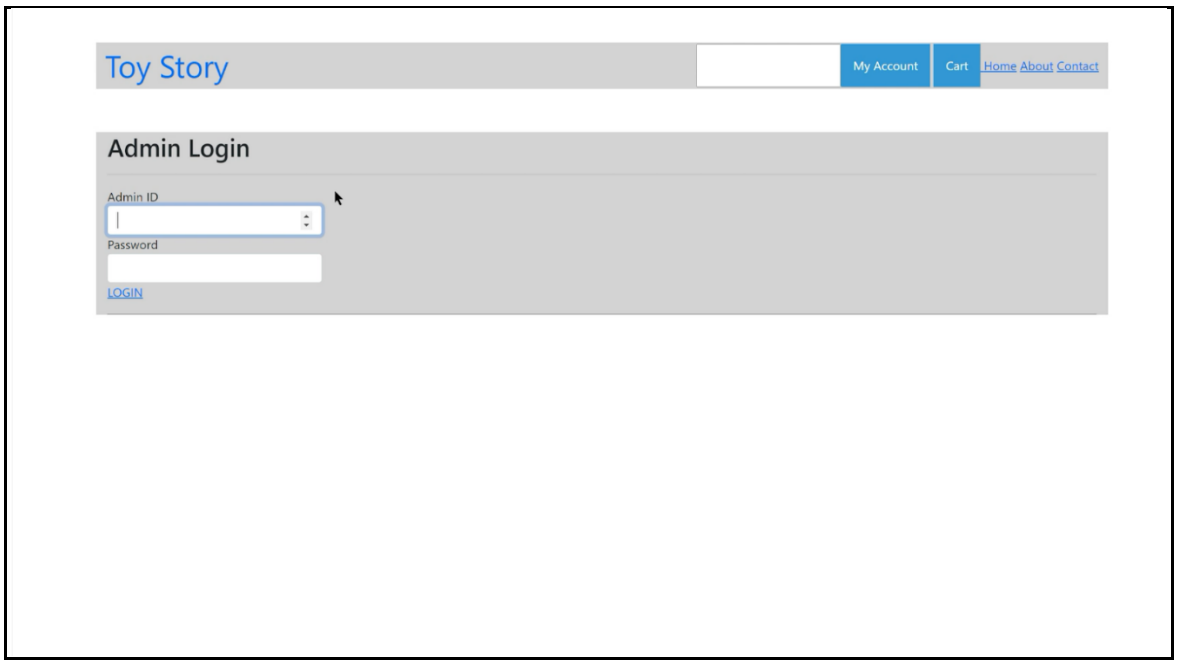
4.2 Login Page:

Login page for both users and administrators.



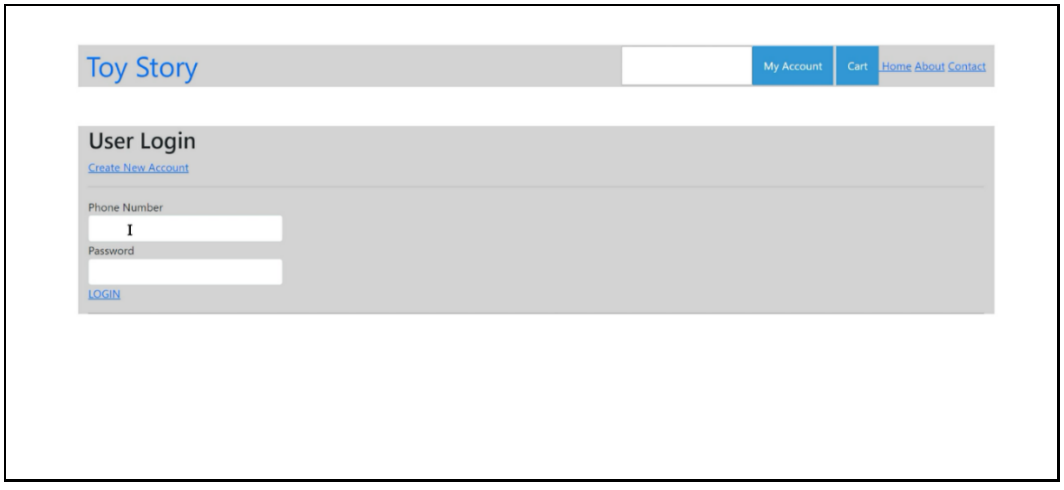
- **Admin Login Page:**

  The difference in the admin and user login page is the role. Here the role is Admin. User and Admin roles will be checked once the page was login and Session will be either Admin or User. The admin needs to use his/her admin Id and password to login into the page. The admin can access the user information. Also, admin can add products, edit product information and add or remove products.



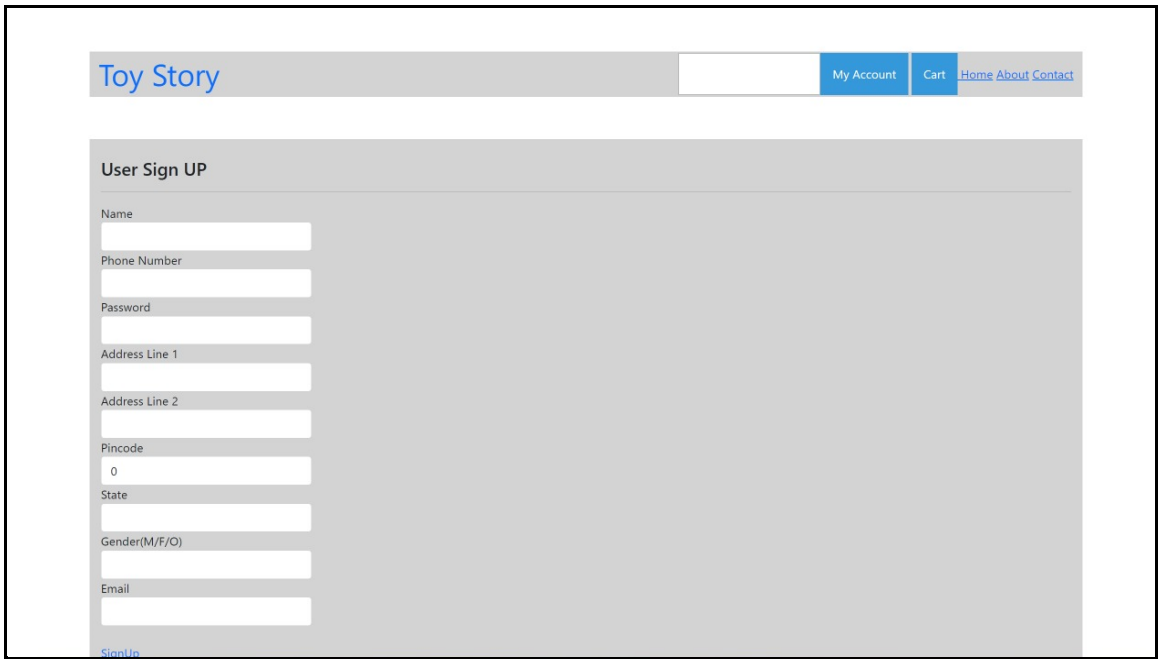
- **User Login Page:**

  In the user login page, the role is User. The user needs to use his/her phone number and password to login into the page. As soon as the user logins into the page he/she will be able to move the products to the cart and make a purchase.



4.3 New User Sign-up Page:

New users can register here. They are required to provide the relevant information that has been asked in order to register on the website to make a purchase.



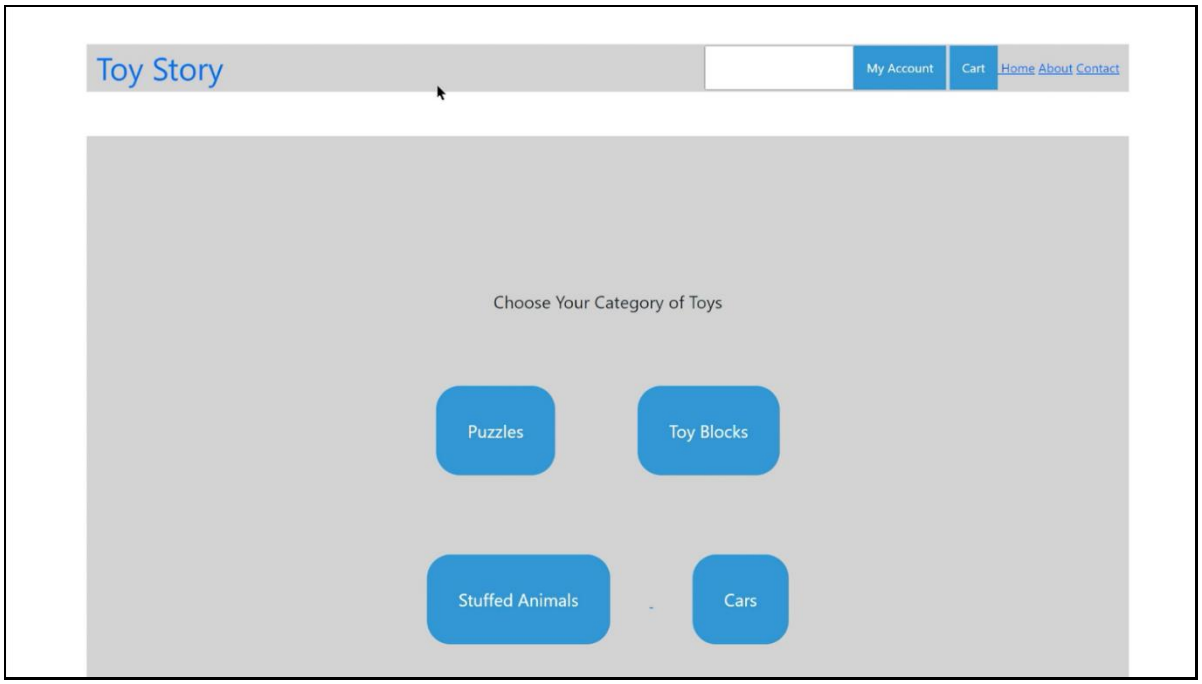
4.4 Admin Home Page:

The admin can view a number of sections here.

- One is the Users Data where all registered users and their information is stored.
- The admin also has the access to add new products, in the Add Products section.
- The Inventory/stock of each product can be viewed, to check availability and update it as well.
- The Sales Data category enables the admin to view all previous orders placed, the billing data and invoice data.

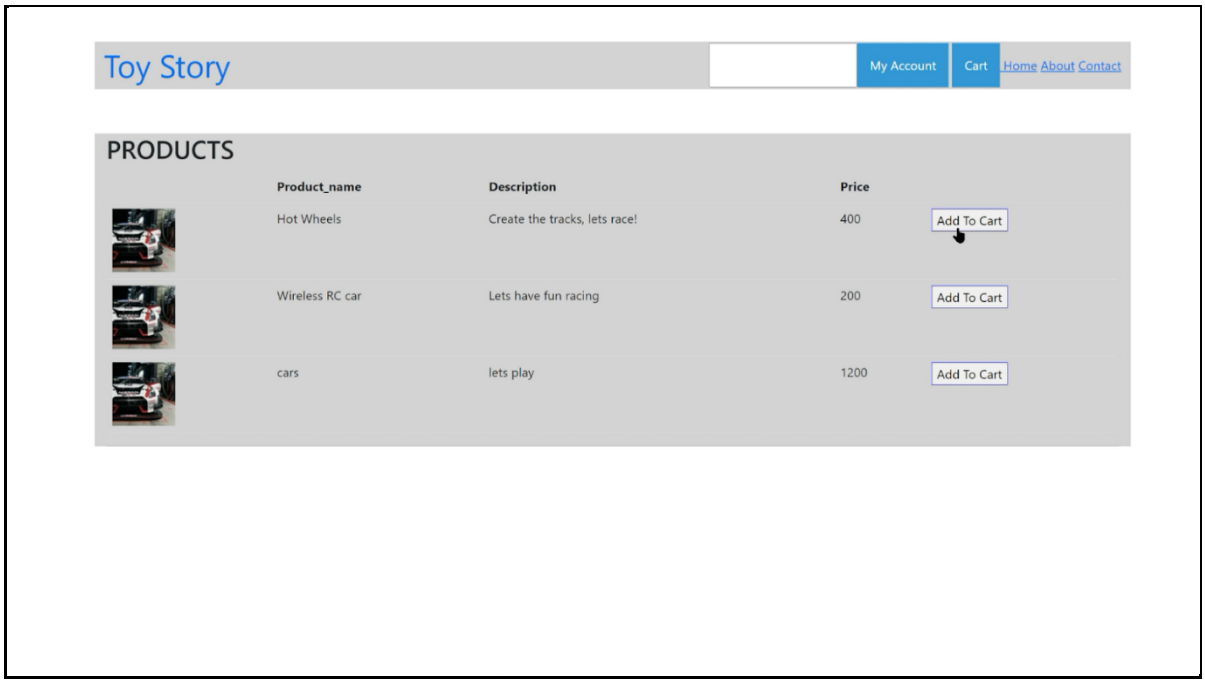
4.5 User Home Page — Category of Products Page:

This page is the Home page of users. It contains all the categories of products that are sold on the website. Choose any category and visit the respective Products page. This page appears the same for both visitors and users.



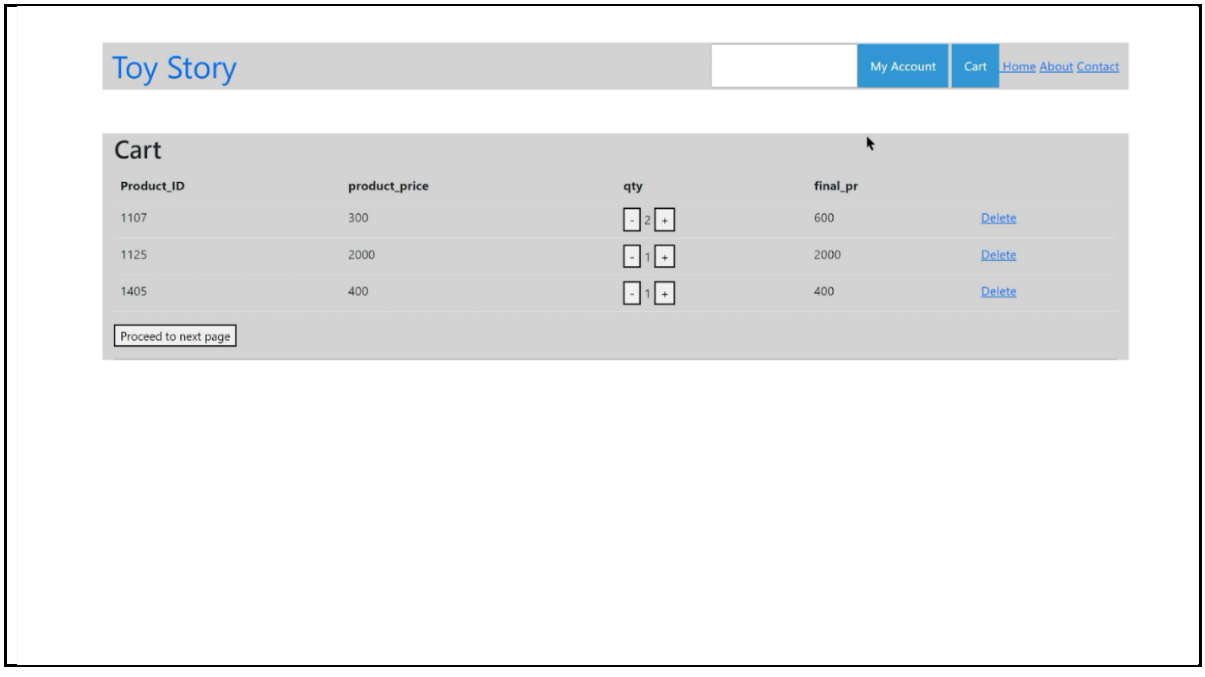
4.6. Products Page:

It consists of different products with image, description, price and Add to cart option. This page appears the same for both visitors and users.



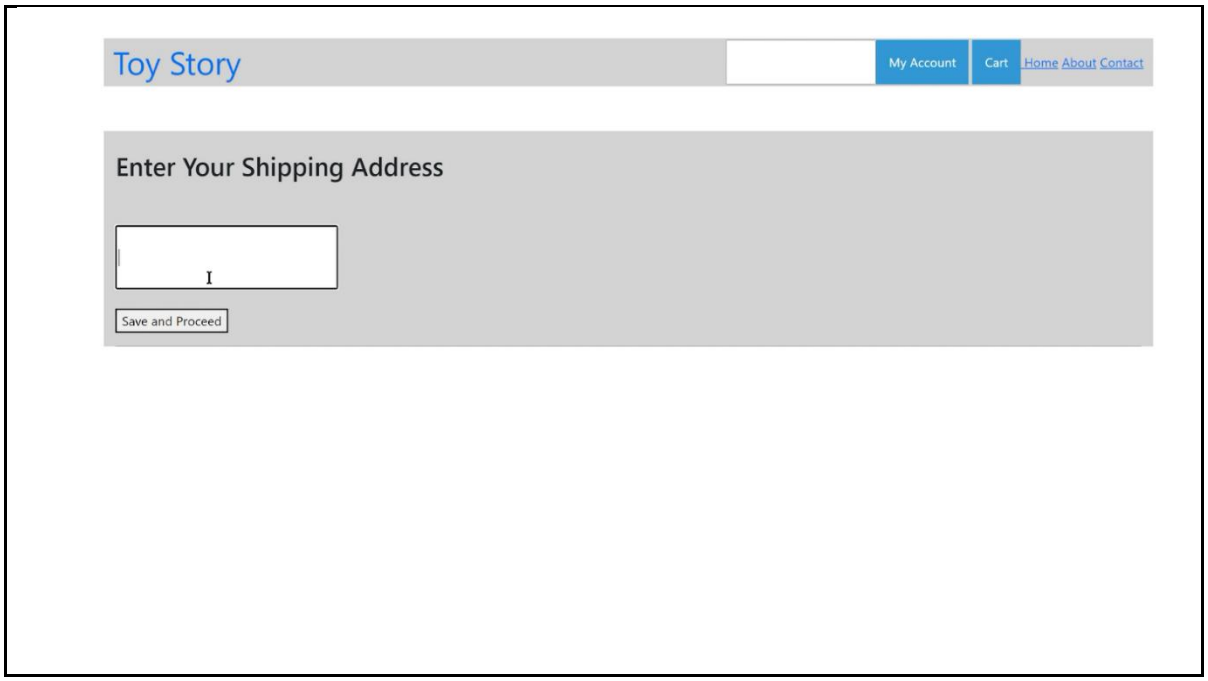
4.7 My Cart:

Once the user adds the item to the cart, they are able to see the selected products and grand total. Users can also edit the quantity of the products as per their requirements, or remove an item too.



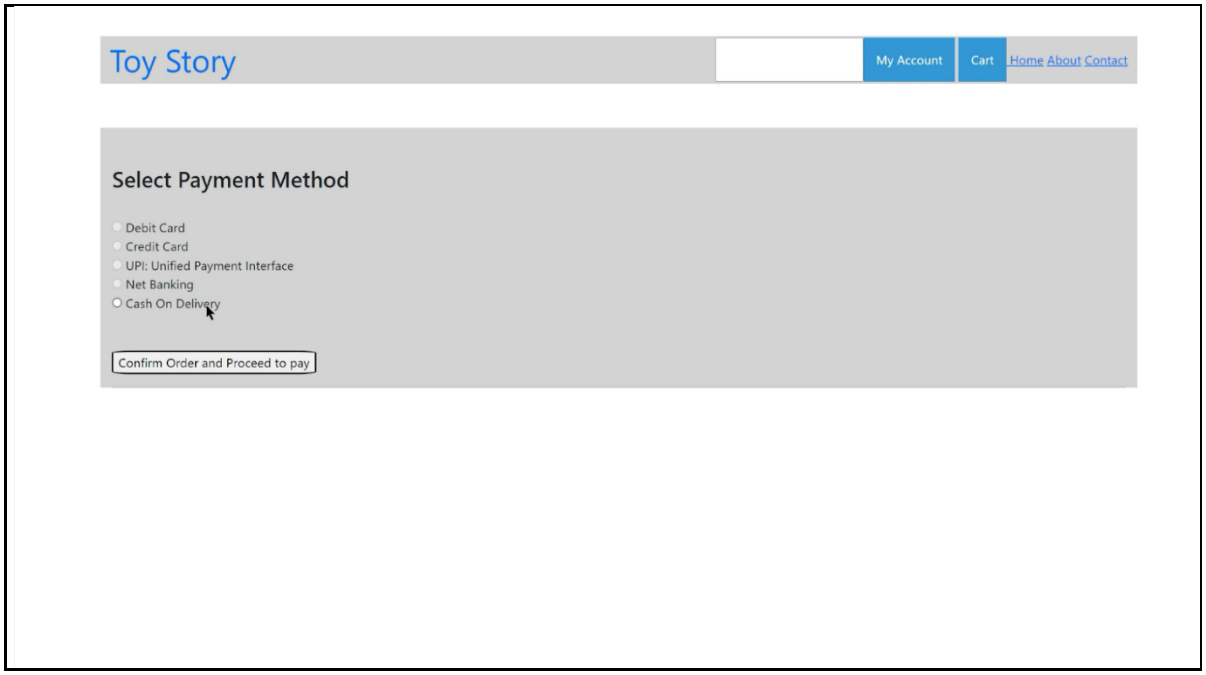
4.8 Billing & Address Page:

Once the user confirms the order of the products they are redirected to enter the shipping address.



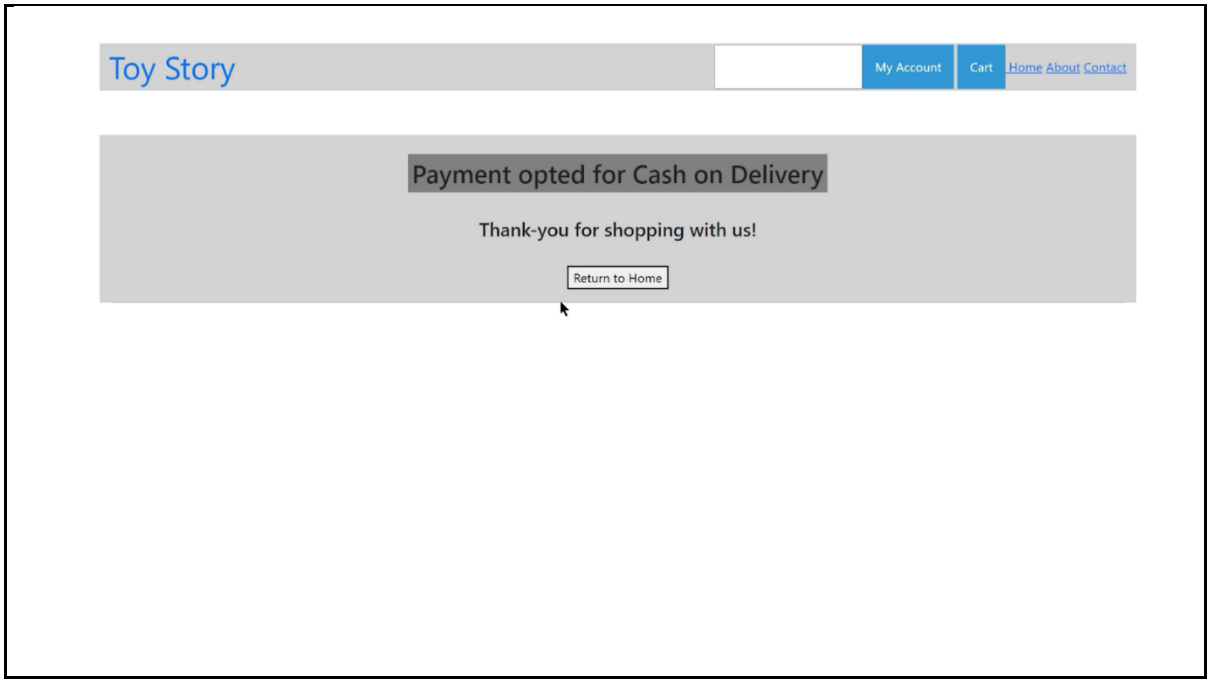
4.9 Transaction Gateway Page:

The users are redirected to the payment page after providing the shipping details.



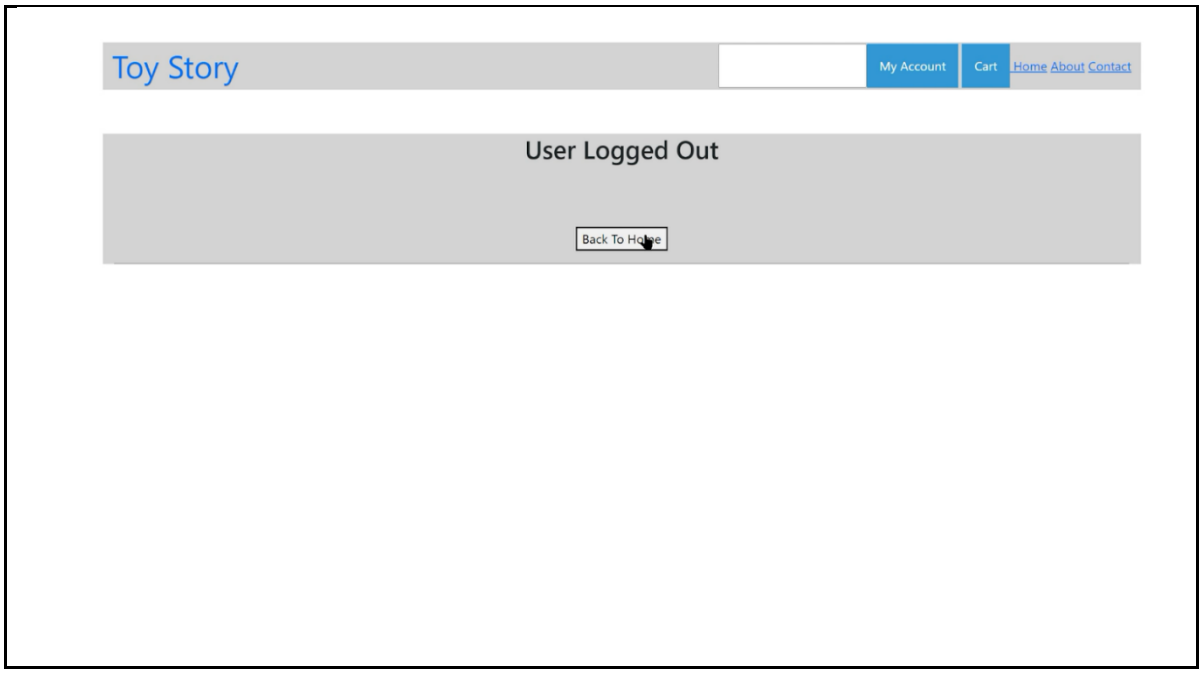
4.10 Payment Confirmation Page:

Once the order is confirmed from the website, the user gets notified.



4.11. Logout Page:

Once the user logs out, the confirmation of the same is shown. And a button “Back to Home” is provided for the user.



5.0 Azure DevOps:

Azure DevOps provides developer services for allowing teams to plan work, collaborate on code development, and build and deploy applications. Azure DevOps supports a collaborative culture and set of processes that bring together developers, project managers, and contributors to develop software.

In Azure DevOps, we have done the following steps

- Logged into dev.azure.com .
- Created the organization as CapstoneProject-G3.
- Created project name as EComSYS-G3.

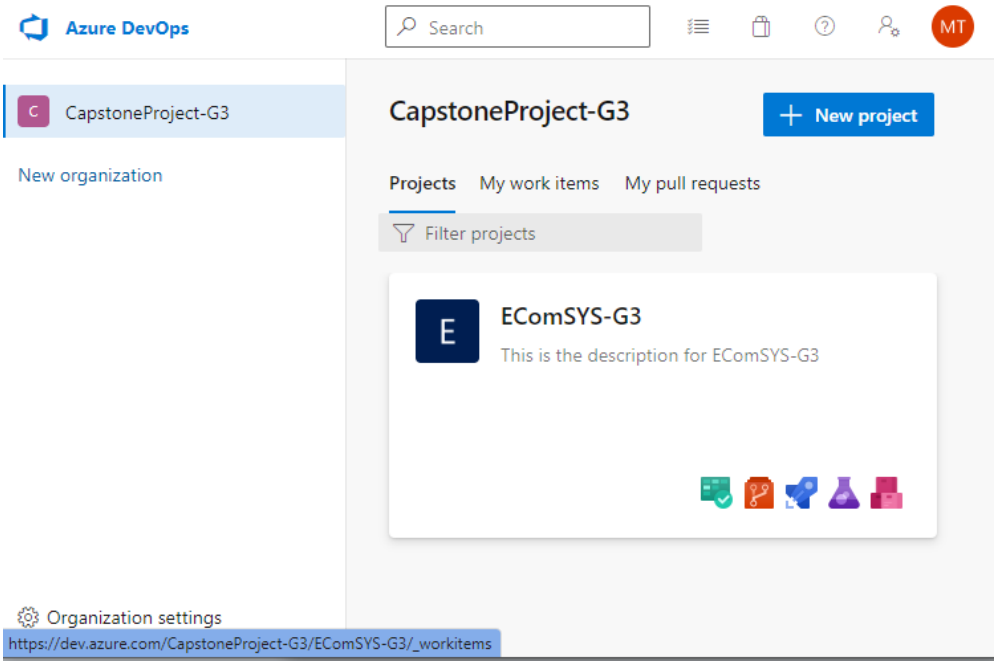


Fig. shows the dev.azure.com, Organization and the project.

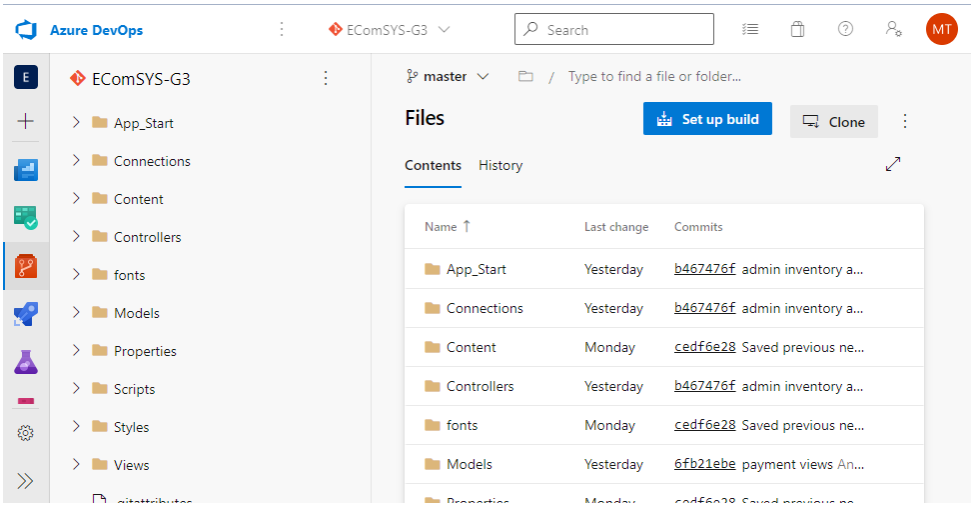


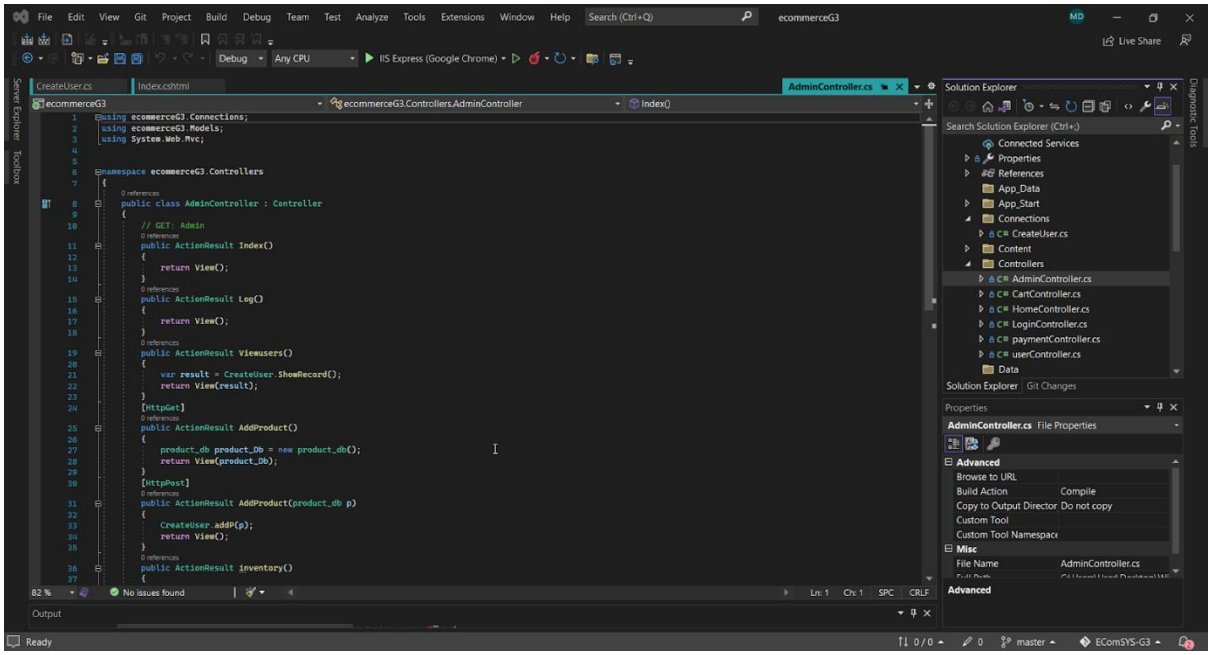
Fig. shows the repo's present in the project.

6.0. Code Snippets:

Controllers:

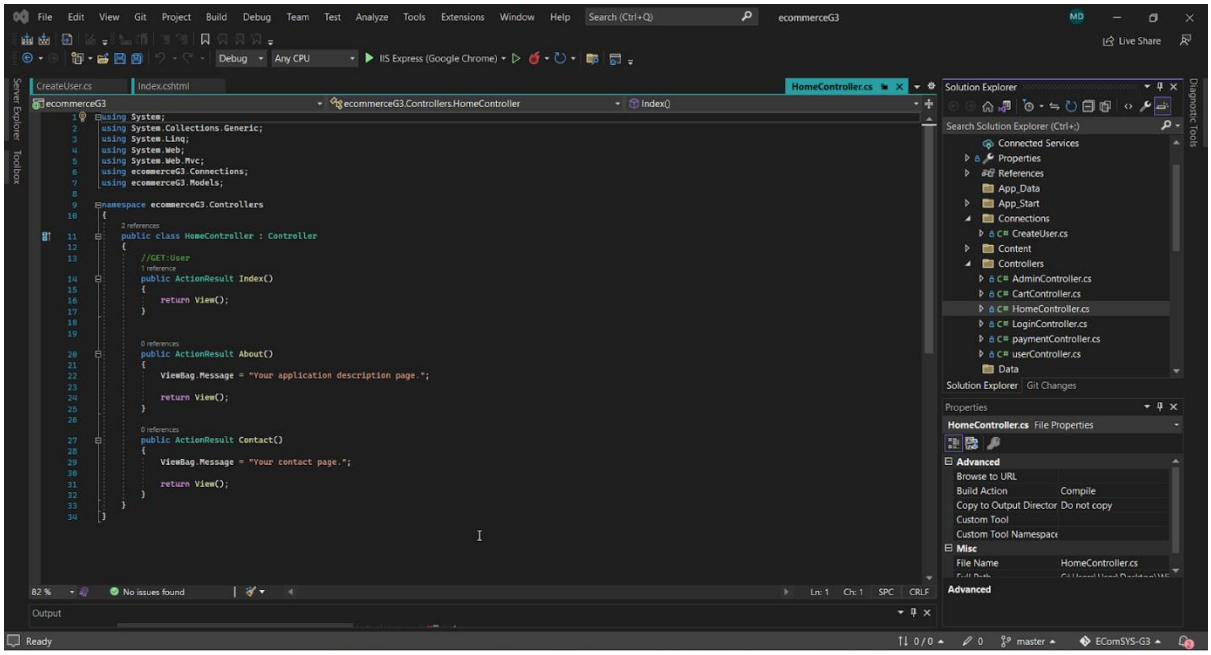
Admin:

```
using ecommerceG3.Connections;
using ecommerceG3.Models;
using System.Web.Mvc;
namespace ecommerceG3.Controllers
{
    public class AdminController : Controller
    {
        // GET: Admin
        public ActionResult Index()
        {
            return View();
        }
        public ActionResult Log()
        {
            return View();
        }
        public ActionResult Viewusers()
        {
            var result = CreateUser.ShowRecord();
            return View(result);
        }
        [HttpGet]
        public ActionResult AddProduct()
        {
            product_db product_Db = new product_db();
            return View(product_Db);
        }
        [HttpPost]
        public ActionResult AddProduct(product_db p)
        {
            CreateUser.addP(p);
            return View();
        }
        public ActionResult inventory()
        {
            var res = CreateUser.showproduct();
            return View(res);
        }
        public ActionResult sales()
        {
            var result = CreateUser.historyview();
            return View(result);
        }
    }
}
```



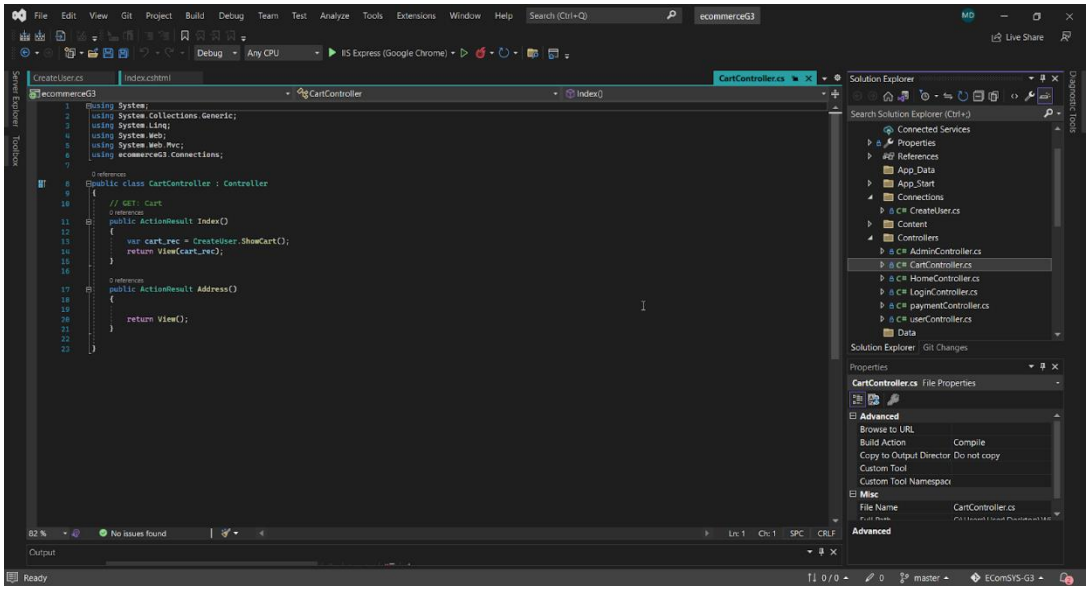
Home:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using ecommerceG3.Connections;
using ecommerceG3.Models;
namespace ecommerceG3.Controllers
{
    public class HomeController : Controller
    {
        //GET:User
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";
            return View();
        }
        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";
            return View();
        }
    }
}
```



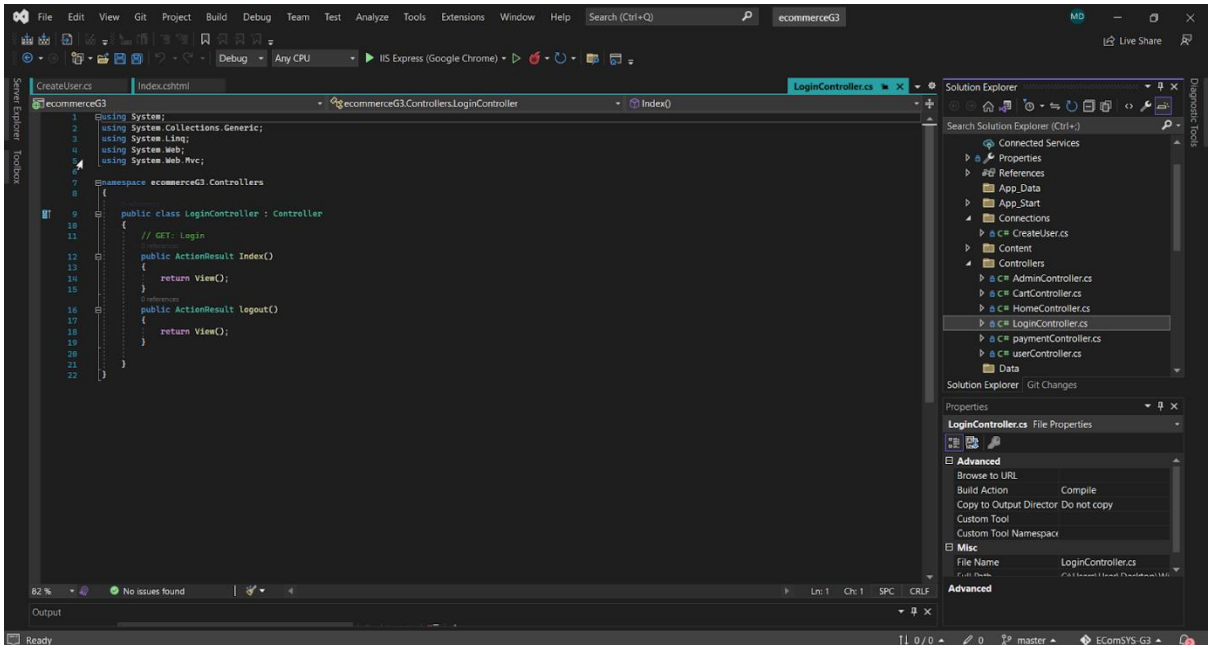
Cart:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using ecommerceG3.Connections;
public class CartController : Controller
{
    // GET: Cart
    public ActionResult Index()
    {
        var cart_rec = CreateUser.ShowCart();
        return View(cart_rec);
    }
    public ActionResult Address()
    {
        return View();
    }
}
```



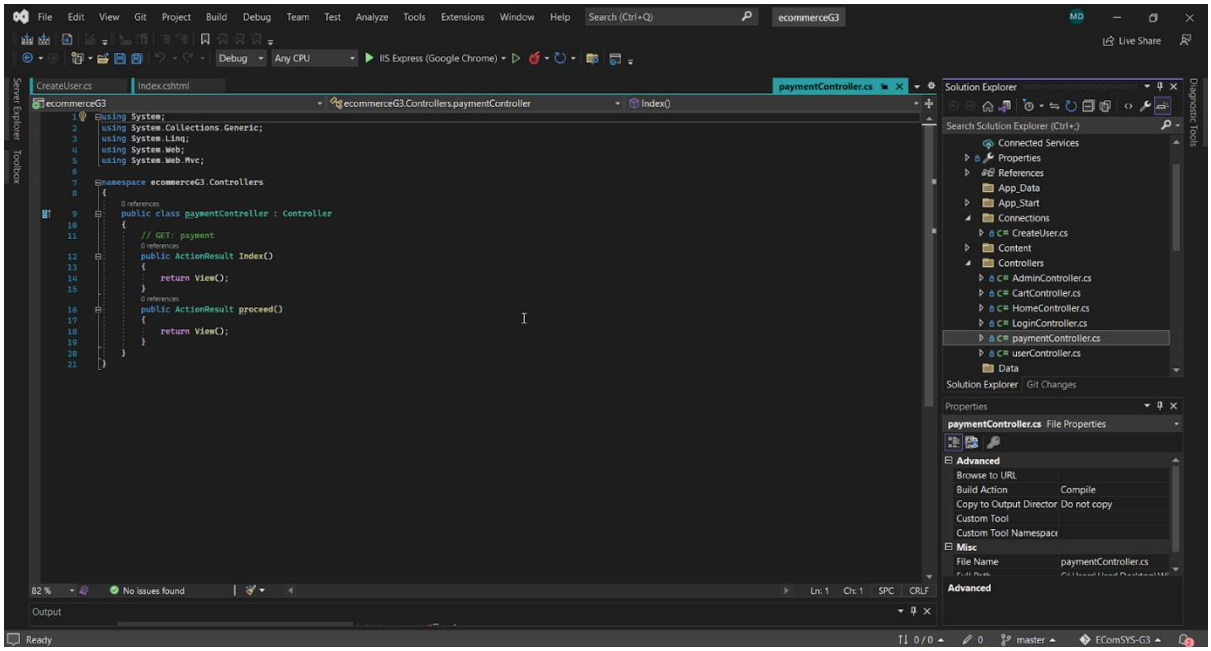
Login:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace ecommerceG3.Controllers
{
    public class LoginController : Controller
    {
        // GET: Login
        public ActionResult Index()
        {
            return View();
        }
        public ActionResult logout()
        {
            return View();
        }

    }
}
```



Payment:

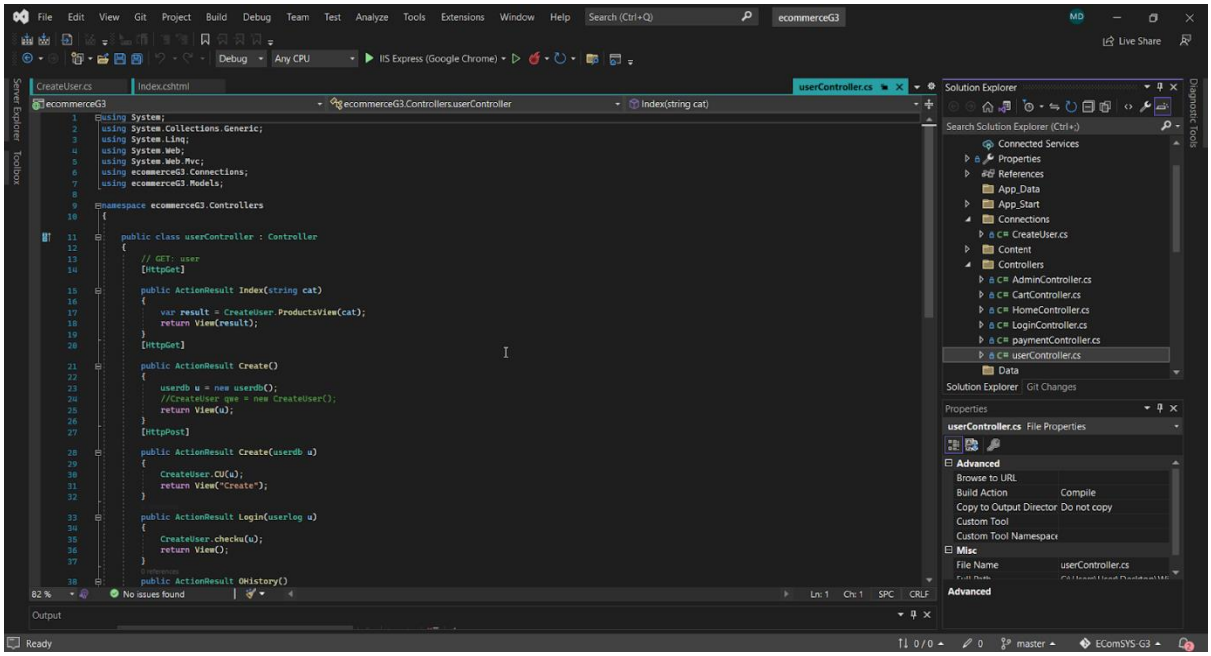
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
namespace ecommerceG3.Controllers
{
    public class paymentController : Controller
    {
        // GET: payment
        public ActionResult Index()
        {
            return View();
        }
        public ActionResult proceed()
        {
            return View();
        }
    }
}
```



User:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using ecommerceG3.Connections;
using ecommerceG3.Models;
namespace ecommerceG3.Controllers
{
    public class userController : Controller
    {
        // GET: user
        [HttpGet]
        public ActionResult Index(string cat)
        {
            var result = CreateUser.ProductsView(cat);
            return View(result);
        }
        [HttpGet]
        public ActionResult Create()
        {
            userdb u = new userdb();
            //CreateUser qwe = new CreateUser();
            return View(u);
        }
        [HttpPost]
        public ActionResult Create(userdb u)
        {
            CreateUser.CU(u);
            return View("Create");
        }
        public ActionResult Login(userlog u)
        {
            CreateUser.checku(u);
            return View();
```

```
        }
        public ActionResult OHistory()
        {
            var result = CreateUser.historyview();
            return View(result);
        }
    }
}
```

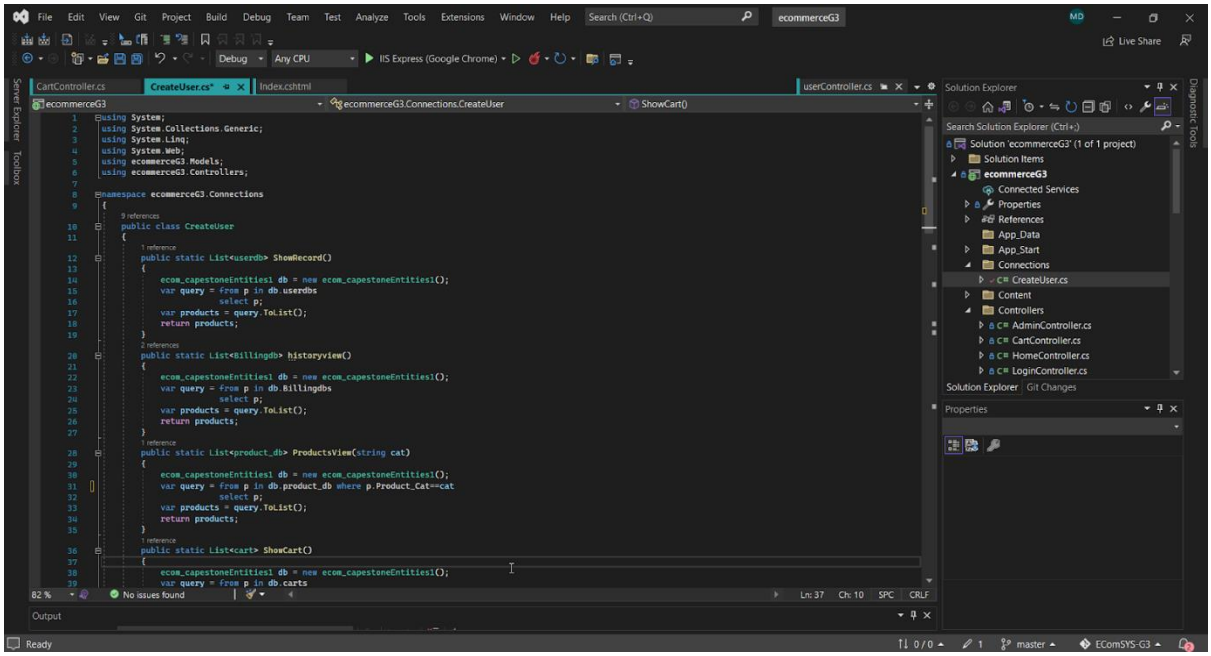


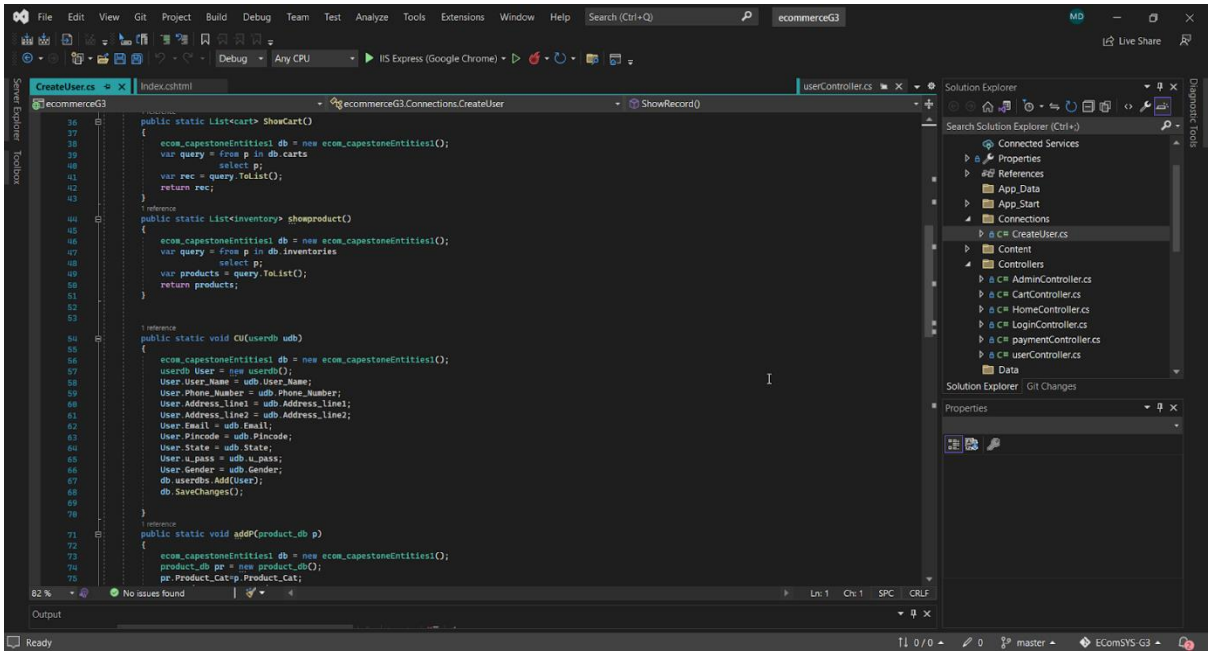
Connections:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using ecommerceG3.Models;
using ecommerceG3.Controllers;

namespace ecommerceG3.Connections
{
    public class CreateUser
    {
        public static List<userdb> ShowRecord()
        {
            ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
            var query = from p in db.userdbs
                        select p;
            var products = query.ToList();
            return products;
        }
        public static List<Billingdb> historyview()
        {
            ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
            var query = from p in db.Billingdbs
                        select p;
```

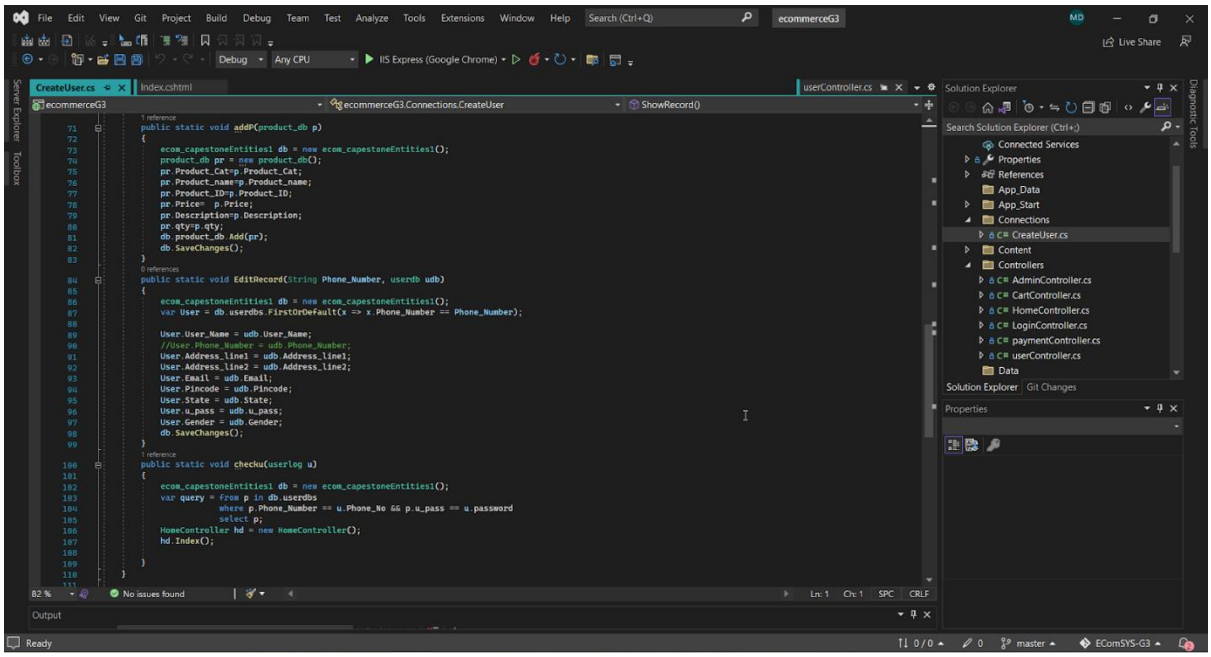
```
            var products = query.ToList();
            return products;
        }
        public static List<product_db> ProductsView(string cat)
        {
            ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
            var query = from p in db.product_db where
p.Product_Cat=="cat"
                        select p;
            var products = query.ToList();
            return products;
        }
        public static List<cart> ShowCart()
        {
            ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
            var query = from p in db.carts
                        select p;
            var rec = query.ToList();
            return rec;
        }
        public static List<inventory> showproduct()
        {
            ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
            var query = from p in db.inventories
                        select p;
            var products = query.ToList();
            return products;
        }
        public static void CU(userdb udb)
        {
            ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
            userdb User = new userdb();
            User.User_Name = udb.User_Name;
            User.Phone_Number = udb.Phone_Number;
            User.Address_line1 = udb.Address_line1;
            User.Address_line2 = udb.Address_line2;
            User.Email = udb.Email;
            User.Pincode = udb.Pincode;
            User.State = udb.State;
            User.u_pass = udb.u_pass;
            User.Gender = udb.Gender;
            db.userdbs.Add(User);
            db.SaveChanges();

        }
        public static void addP(product_db p)
        {
            ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
            product_db pr = new product_db();
            pr.Product_Cat=p.Product_Cat;
            pr.Product_name=p.Product_name;
            pr.Product_ID=p.Product_ID;
            pr.Price=  p.Price;
            pr.Description=p.Description;
            pr.qty=p.qty;
            db.product_db.Add(pr);
            db.SaveChanges();
        }
        public static void EditRecord(String Phone_Number, userdb udb)
```

```
        {
            ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
            var User = db.userdbs.FirstOrDefault(x => x.Phone_Number ==
Phone_Number);

            User.User_Name = udb.User_Name;
            //User.Phone_Number = udb.Phone_Number;
            User.Address_line1 = udb.Address_line1;
            User.Address_line2 = udb.Address_line2;
            User.Email = udb.Email;
            User.Pincode = udb.Pincode;
            User.State = udb.State;
            User.u_pass = udb.u_pass;
            User.Gender = udb.Gender;
            db.SaveChanges();
        }
        public static void checku(userlog u)
        {
            ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
            var query = from p in db.userdbs
                        where p.Phone_Number == u.Phone_No && p.u_pass ==
u.password
                        select p;
            HomeController hd = new HomeController();
            hd.Index();
        }
    }
}
```





```
public static List<cart> ShowCart()
{
    ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
    var query = from p in db.carts
                select p;
    var rec = query.ToList();
    return rec;
}
public static List<inventory> showproduct()
{
    ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
    var query = from p in db.inventories
                select p;
    var products = query.ToList();
    return products;
}

public static void CU(userdb udb)
{
    ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
    userdb User = new userdb();
    User.User_Name = udb.User_Name;
    User.Phone_Number = udb.Phone_Number;
    User.Address_line1 = udb.Address_line1;
    User.Address_line2 = udb.Address_line2;
    User.Email = udb.Email;
    User.Pincode = udb.Pincode;
    User.State = udb.State;
    User.u_pass = udb.u_pass;
    User.Gender = udb.Gender;
    db.userdbs.Add(User);
    db.SaveChanges();

}
public static void addP(product_db p)
{
    ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
    product_db pr = new product_db();
    pr.Product_Cat=p.Product_Cat;
```



```
public static void addP(product_db p)
{
    ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
    product_db pr = new product_db();
    pr.Product_Cat=p.Product_Cat;
    pr.Product_name=p.Product_name;
    pr.Product_ID=p.Product_ID;
    pr.Price=  p.Price;
    pr.Description=p.Description;
    pr.qty=p.qty;
    db.product_db.Add(pr);
    db.SaveChanges();
}
0 references
public static void EditRecord(String Phone_Number, userdb udb)
{
    ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
    var User = db.userdbs.FirstOrDefault(x => x.Phone_Number == Phone_Number);

    User.User_Name = udb.User_Name;
    //User.Phone_Number = udb.Phone_Number;
    User.Address_line1 = udb.Address_line1;
    User.Address_line2 = udb.Address_line2;
    User.Email = udb.Email;
    User.Pincode = udb.Pincode;
    User.State = udb.State;
    User.u_pass = udb.u_pass;
    User.Gender = udb.Gender;
    db.SaveChanges();
}
public static void checku(userlog u)
{
    ecom_capestoneEntities1 db = new ecom_capestoneEntities1();
    var query = from p in db.userdbs
                where p.Phone_Number == u.Phone_No && p.u_pass == u.password
                select p;
    HomeController hd = new HomeController();
    hd.Index();

}
}
```

7.0 Conclusion:

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur's but also from the customer's point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible.

As per a survey, most consumers of online stores are impulsive and usually make a decision to stay on a site within the first few seconds. "Website design is like a shop interior. If the shop looks poor or like hundreds of other shops the customer is most likely to skip to the other site. Hence we have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible. In this project, the user is provided with an ecommerce web site that can be used to buy books online. To implement this as a web application we used ASP.NET as the Technology. ASP.NET has several advantages such as enhanced performance, scalability, built-in security and simplicity.

To build any web application using ASP.NET core MVC. We need a programming language such as C#, VB.NET, J# and so on. C# was the language used to build this application. For the client browser to connect to the ASP.NET engine we used Microsoft's Internet Information Services (IIS) as the Web Server. Asp.NET Core Web API with Entity Framework Database first approach to interact with the database as it provides in-memory caching that eliminates the need to contact the database server frequently and it can easily deploy and maintain an ASP.NET application. SQL was used as a back-end database since it is one of the most popular databases, and it provides fast data access, easy installation and simplicity.

A good shopping cart design must be accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart application described in this project provides a number of features that are designed to make the customer more comfortable.

This project helps in understanding the creation of an interactive web page and the technologies used to implement it. The design of the project which includes Data Model and Process Model illustrates how the database is built with different tables, how the data is accessed and processed from the tables. The building of the project has given me a precise knowledge about how ASP.NET Core MVC is used to develop a website, how it connects to the database to access the data and how the data and web pages are modified to provide the user with a shopping cart application.