

Edward Barrier III

edwardbarrieriii@outlook.com | <http://www.edwardbarrier.com/>



KC Digital

Reservation System

About me

Growing up, I've always been a curious person. It started with RC cars when I was young, I loved seeing all of the small components build one big piece that worked. Moving on from RC cars to computers in my teenage years, I was doing practically the same thing just with different pieces and more of a challenge. Now moving on from building computers into software development, I find the same sense of fulfillment and excitement when building an application.

I discovered web development in college when I needed a computer science class for my degree. I enrolled in a web development course where we learned HTML, CSS, and some basic principles of development. I enjoyed it enough to take the advanced course where we learned JavaScript. After these courses I couldn't take any more computer science classes for a couple years until I transferred to the University of Central Missouri.

This is where I decided to drop out of college and enroll into Centriq. I felt as if I was paying for classes that didn't benefit my future in college. Whereas in Centriq, I feel like I'm getting what I paid for.

Project Brief

KC Digital is a computer repair shop that has multiple locations that needs an online application that will allow their clients to register themselves, add their computers, and schedule a reservation to get them into the shop. Clients should be able to schedule at any location, however, each location has a limit to the number of reservations it can take each day. If the location is at capacity for the day, clients should not be able to create a reservation. However, company administrators should be able to override the limit to manually put one in.

Use Case: There should be three different roles with their own capabilities.

| | | |
|------------|-------------|---|
| - Customer | Name | |
| - Employee | Admin | Edit Details Delete |
| - Admin | Customer | Edit Details Delete |
| | Employee | Edit Details Delete |

Use Case

| | | | | | |
|-------------------------------|-------|---------|---------|------|--------|
| HomeController | Index | Contact | | | |
| Customer | x | x | | | |
| Employee | x | x | | | |
| Admin | x | x | | | |
| ComputersController | Index | Create | Details | Edit | Delete |
| Customer | x | x | x | x | x |
| Employee | x | | | x | |
| Admin | x | | | x | |
| LocationsController | Index | Create | Details | Edit | Delete |
| Customer | x | | | | |
| Employee | x | | | | |
| Admin | x | x | x | x | x |
| ReservationsController | Index | Create | Details | Edit | Delete |
| Customer | x | x | x | x | x |
| Employee | x | | | | |
| Admin | x | x | x | x | x |

Administration Capabilities:

- CRUD functionality for Company Locations
- View all reservations
- Override location limits and place manual reservations
- Add/update notes

- These snippets show different functionalities of the users and how I organized them.

Employee Capabilities:

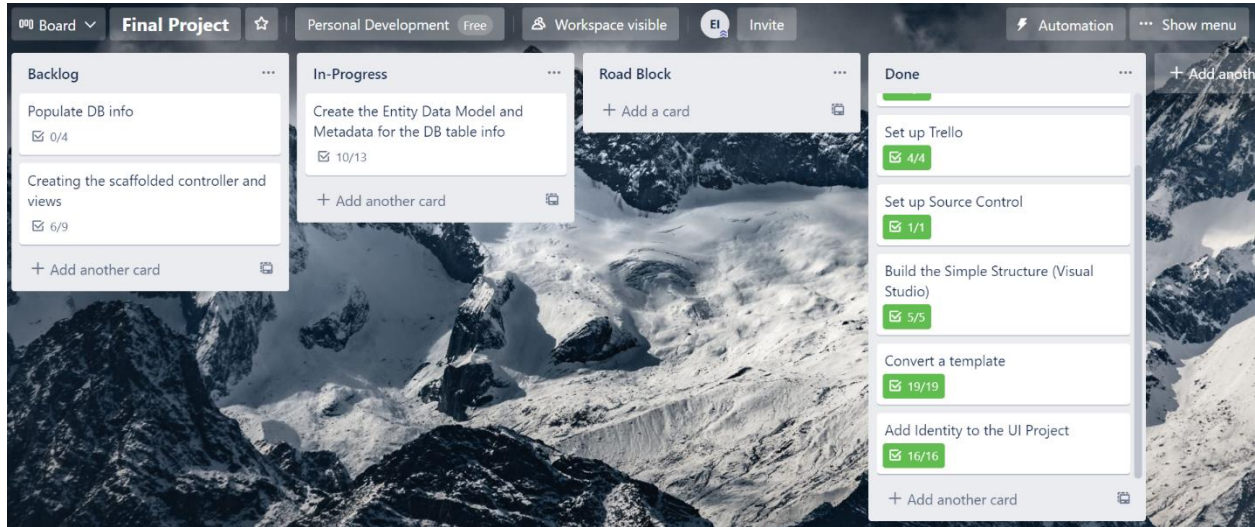
- View all reservations by location and notes
- Add/update notes

Owner Capabilities:

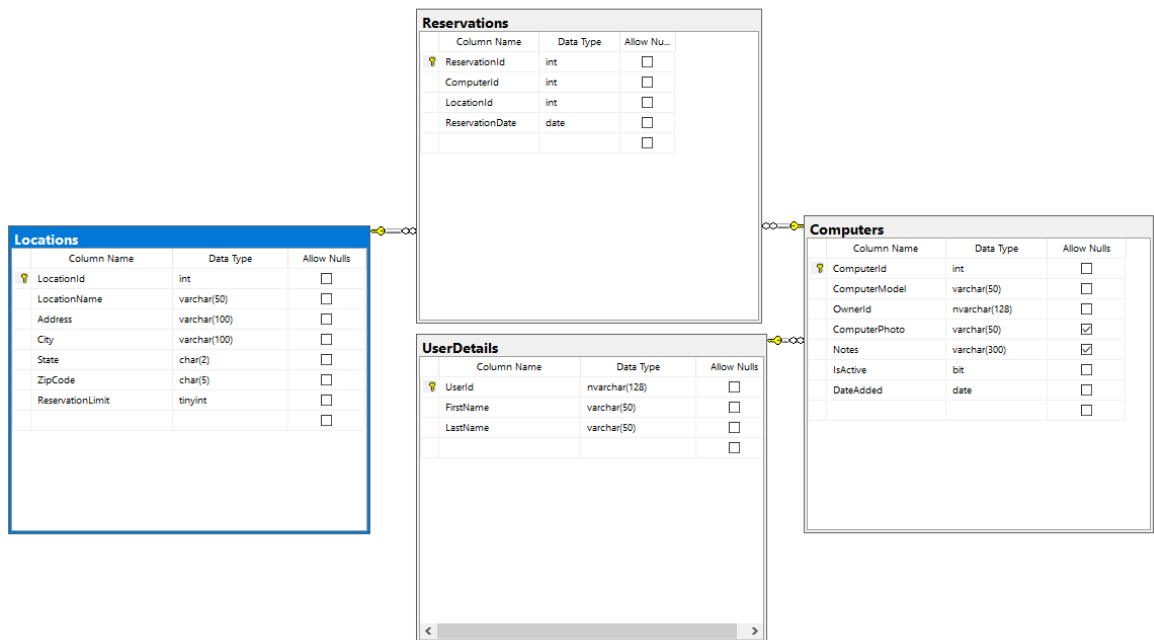
- CRUD functionality for their account
- CRUD functionality for their asset profile(s)
- Upload a photo for assets
- Add/update notes to their asset profile
- Place reservations for assets
- View their own asset profiles and reservations

Agile

Scrum Methodology



Database Schema



User Experience

The user should only be able to see their own computers.

```
// GET: Computers
[Authorize(Roles = "Admin, Customer, Employee")]
public ActionResult Index()
{
    var currentUser = User.Identity.GetUserId();
    if (User.IsInRole("Customer"))
    {
        var computers = db.Computers.Where(c => c.OwnerId == currentUser).Include(c => c.UserDetail).ToList();
        return View(computers.ToList());
    }
    return View(db.Computers.ToList());
}
```

This snippet shows the code written to achieve that. This logic is also applied in the Computers/Create and Computers/Edit ActionResults.

File Upload

- Customers should be able to add pictures of their computers for 1. If there is any external damage on them we can see. 2. If there is several models in our shop, there's not a possibility of getting them mixed up.

Made sure to change the DisplayFor to an image so it will output the image instead of the image filename.

```
</td>
<td>
    @* @Html.DisplayFor(modelItem => item.ComputerPhoto) *
    
</td>
```

Then added the controller logic.

```

#region File Upload
if (ModelState.IsValid)
{
    //use default img if none is provided
    string file = "noImg.png";
    if (logo != null)
    {
        file = logo.FileName;
        string ext = file.Substring(file.LastIndexOf("."));
        string[] goodExts = { ".jpeg", ".jpg", ".png", ".gif" };
        if (goodExts.Contains(ext.ToLower()) && logo.ContentLength <= 4194304)
        {
            file = Guid.NewGuid() + ext;

            #region Resize Image
            string savePath = Server.MapPath("~/Content/images/computers/");

            Image convertedImage = Image.FromStream(logo.InputStream);

            int maxImageSize = 500;

            int maxThumbSize = 100;

            ImageUtility.ResizeImage(savePath, file, convertedImage, maxImageSize, maxThumbSize);

            #endregion
        }
        computer.ComputerPhoto = file;
    }
}
#endregion

```

Reservation Limit

If the reservation limit is reached on a certain day, it shouldn't let the customer schedule a reservation on that day. Unless, the user is in the role of "Admin"

```

[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "ReservationId,ComputerId,LocationId,ReservationDate")] Reservation reservation)
{
    if (ModelState.IsValid)
    {
        var location = db.Locations.Where(l => l.LocationId == reservation.LocationId).FirstOrDefault();
        //Setting a reservation limit, if the limit is reached, send the user to an error page.
        if (location.Reservations.Where(r => r.ReservationDate == reservation.ReservationDate).ToList().Count >= location.ReservationLimit && !User.IsInRole("Admin"))
        {
            return RedirectToAction("Index", "Errors");
        }
        db.Reservations.Add(reservation);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
}

```

- This logic also applies in the Reservations/Edit

EDWARD BARRIER III

edwardbarrieriii@outlook.com · (816) 520-3058 · www.edwardbarrier.com

LinkedIn: www.linkedin.com/in/EdwardBarrierIII

FULL STACK DEVELOPER

.NET Framework
Full Stack Web Application Design
C# Fundamentals
Database Management

Unit Testing
Software Development Lifecycle
Agile Scrum Methodology
Continued Growth in IT Knowledge

TECHNICAL AND PROFESSIONAL SKILLS

Front End: HTML5, JavaScript, jQuery, jQueryUI, CSS3, Responsive/Mobile Web Development, Bootstrap, ReactJS

Middle Tier: Visual Studio, C#.NET, ASP.NET, LINQ, MVC, EF

Back End: ADO.NET, SQL, SQL Server, SSMSE

Professional Skills: Troubleshooting, Critical Thinking, Communication, Project Management Fundamentals, Teamwork

INDEPENDENT DEVELOPMENT PROJECTS

- **Personal Site:** Created an application to display information about myself, examples of my work, testimonials, contact information, and my current resumé. I have also included a section for my team members because they have played a huge part in my education. For this I used Bootstrap, HTML5, CSS3, and JavaScript/jQuery.
- **Dungeon Application:** Created a simple game using C#. Including character selection and weapon selection, the player walks through dungeons and battles monsters until he/she dies. The application then displays the players score and the game ends.
- **Front-End Development Project:** Created an application that follows a start-up company's project brief. This application is built with HTML5, CSS3, and JavaScript/jQuery.
- **Rust Random Task Generator:** Created an application to give me random tasks on my favorite game. The main purpose for this application was to dive deeper into JavaScript/jQuery functionality. This is built with HTML5, CSS3, and JavaScript/jQuery.
- **StoreFront:** Created a secure application for managing product data. Application is built to simulate an online store front with a shopping cart. Administrators have the ability to manage product, category and vendor data.

- **S.A.T. Scheduling Administration Tool:** Created a secure application for managing product data. Application is built to simulate an online class scheduling system. Administrators will have the ability to manage students, courses, scheduled classes, and enrollments.
 - **Final Project:** Created a secure data-driven ASP.NET MVC application from design through deployment for managing the tracking and organization of hardware and software within a company. Administrators have the ability to manage employee, department data and all details relating to assigned hardware and software.
-

TECHNICAL TRAINING

Full Stack Web Developer Program, Centriq Training
City, State

Anticipated Graduation 06/2021

- MVC Framework
 - Trouble Shooting & Debugging
 - Source Control
 - Agile/Scrum (Created Team Project)
 - Website Deployment
 - Pair Programming
 - Code Review
-

WORK EXPERIENCE

Package Handler, FedEx Ground
Kansas City, Missouri

12/2020 – 04/2021

- Stored items in an orderly and accessible manner in three trucks at a time.
- Developed constructive and cooperative working relationships with others and maintained them over time.
- Managed and scanned 100% of the packages that I worked with.

Sales Advisor, Office Depot
Independence, Missouri

02/2019-12/2020

- Recommended, selected, and helped locate or obtain merchandise based on customer needs and desires.
- Designed flyers, posters, postcards, etc. for 20+ customers a week.
- Gained knowledge of principles and methods for showing, promoting, and selling products or services. This included marketing strategy and tactics, product demonstration, sales techniques, and sales control systems.
- Took initiative to learn various positions, such as printer/computer sales and cash register, in order to assist where needed most.

Lube Technician, Adams Toyota
Lee's Summit, Missouri

09/2017-11/2018

- Inspected vehicles for damage and recorded findings so that necessary repairs could be made.
- Performed routine and scheduled maintenance services, such as oil changes, lubrications, and tune-ups.
- Obtained knowledge of machines and tools, including their designs, uses, repair, and maintenance.
- Analyzed information and evaluated results to choose the best solution and solve problems.

