

Jordan Hickman

RESERVATION SYSTEM

About Me

To say that my experience at Centriq Training up to now has been exciting and life changing would be an understatement. I have learned so much more about myself as well as the .NET Full Stack Development umbrella. I have reconstructed my inner patience for myself learning something foreign to me. I have learned to harness a new pace and determination while striving to digest new skills. I spoke with industry peers, fellow students, teachers of all kinds, and veteran freelancers and they have all told me the same thing. To solidify my newfound depth, I must create a routine. Something I have not been exactly disciplined in my entire life. I can now feel and visualize this.

Why Technology?

Upon choosing technology, I must understand I grew up at a time that technology was at a quickened growth rate and at a time where web technology was reaching its highest high. I have used technology at an early age, and we were taught computer classes in public school. We do not always land on technology as a career path despite that fact. I realized that I wanted to choose technology as a career path during my mid to late career path in Specialty Coffee. We take technology for granted and they make all work and life as we know it more enjoyable and easier to work with.

Why Coding? Why Reservation System?

Coding is a vehicle in which you can make sure you ride the wave of making life changing technologies while being encompassed in a growing industry and a supportive community. That is why I chose this path.

My favorite part about coding is that it is seemingly the primordial soup or base for how a lot of technology is created. For example, the problems we can solve with coding are growing every day. In the context of my project "Reservation System" I am solving the problem of a music studio trying to have new clients able to book time slots from their home or phone. I solved that problem. I chose this project because being a musician myself, I have been in and out of studios to record a handful of times and the way in which you book at the studios seemed very archaic to me. I want to aid in the regulation of solo entrepreneurs using web apps.

Details

Presentation Date: June 17th, 2021
Kansas City, Missouri
816-820-3938
jdickman816@outlook.com

Technologies

HTML

CSS

JavaScript

jQuery

C#

ASP.NET MVC

SQL

Trello

Creality

Project Requirements

The company has multiple locations and needs an online application that will allow clients to register themselves and set up a reservation. Clients can schedule at any location, however; each location has a limit to the number of reservations that it can take at one time. If the location is at capacity clients should not be able to create a reservation. Company Administrators, however, can override the limit and manually enter a reservation.

Target Audience

Clients, staff, owners / administration.

Admin Capabilities

- ❖ CRUD functionality for Company Locations.
- ❖ View all reservations made.
- ❖ Override location limits and place manual reservations.
- ❖ Add/Update notes.

Employee Capabilities

- ❖ View all reservations by location and notes.
- ❖ Add/Update notes.

Owner Capabilities

- ❖ CRUD functionality for their account.
- ❖ CRUD functionality for their asset profiles.
- ❖ Upload a photo for assets.
- ❖ Add/Update notes to their asset profile.
- ❖ Place reservations for assets.
- ❖ View their own asset profiles and reservations.

Branding

This is an external application. The client is open to any branding and template options. Just needs a modern and consistent feel throughout.

Insight

Their clientele are very busy people and most of their web traffic is through mobile devices. Eye appealing design and ease of use for mobile devices is a must.

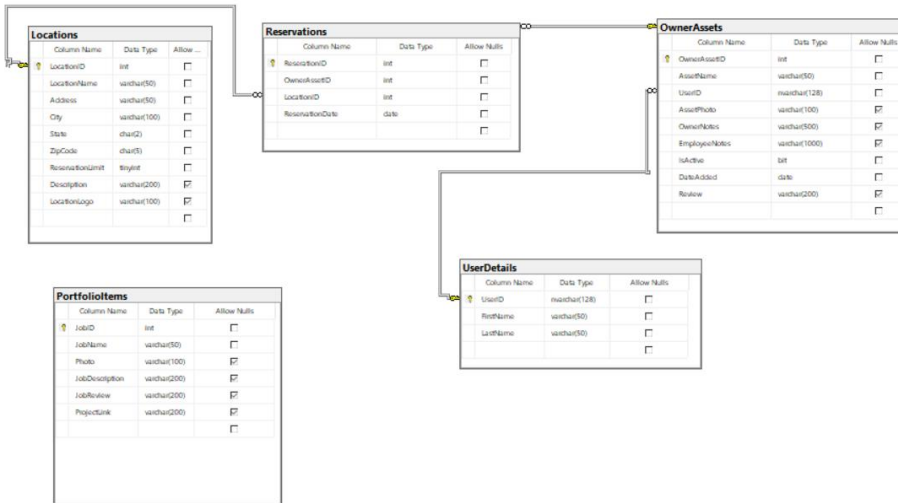


Figure 1: SQL Database Diagram

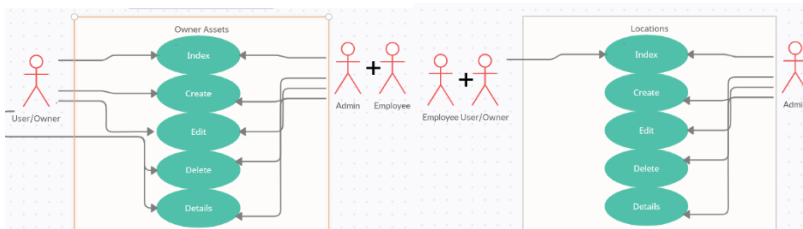


Figure 2: Owner Asset Use/Case

Figure 3: Location Use/Case

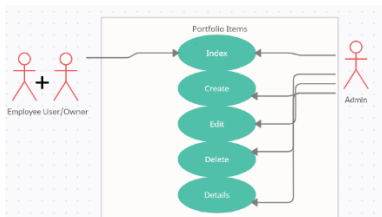


Figure 4: Portfolio Items Use/Case

Database Diagram (SQL) and Use/Case Diagrams

Figure 1: This is the Database Diagram from SQL. I included this as a visual to describe the foreign key / primary key placements and the data types used for the data that's being created and stored.

Figure 2-4: I included these plan schemes representing the user's profiles and their roles. My client, the studio owner or 'Admin'. The studio owner's employees and last the studio's client profiles. These diagrams help visualize what I was giving which role access to the data.

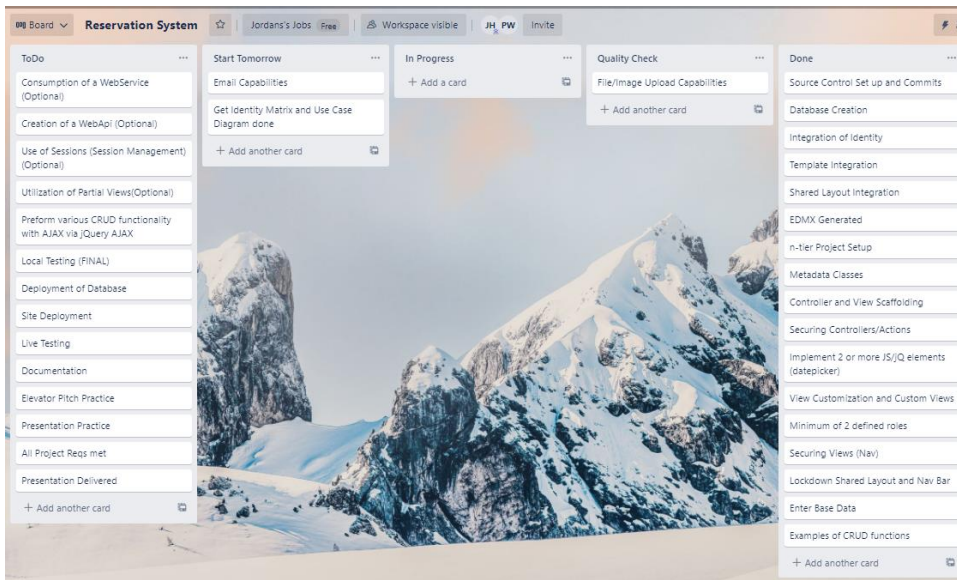
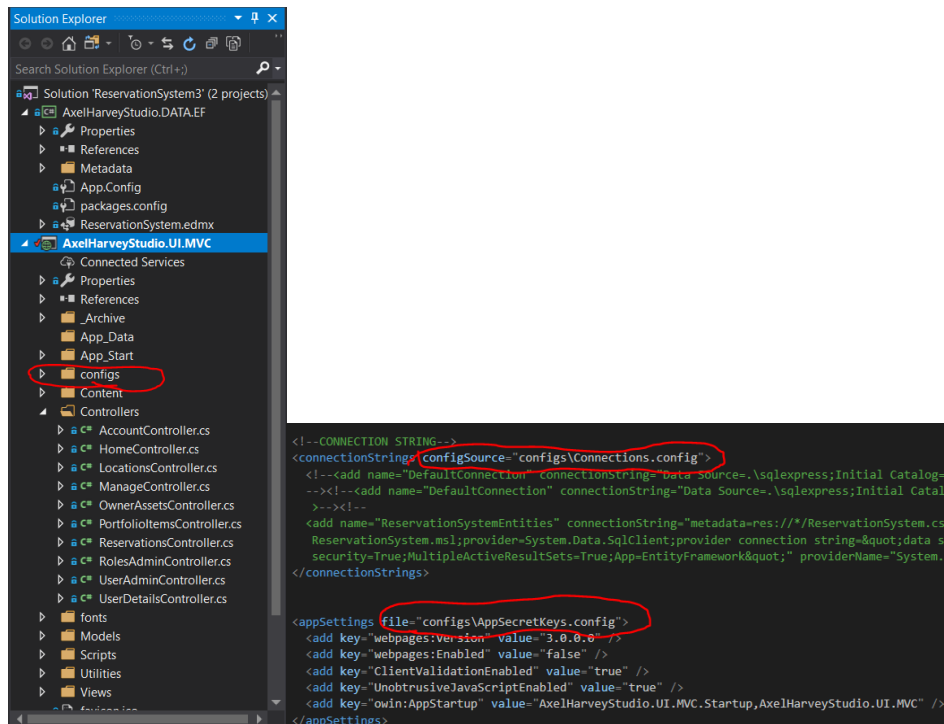


Figure 5: Kanban Board with Trello

Trello Kanban Board

Figure 5: This is a look into the planning process for each chunk or block of project completion. I had a main 'To Do' list, 'Start Tomorrow', 'In Progress', 'Quality Check', and 'Done' list. I would start adding steps to the 'To Do' and when I would start on 1 task I would move that to 'In Progress'. When it was complete I would first move it to the 'Quality Check' list. As I moved on to the next 'To Do' list task, I would wait until about 4 or 5 items made it to the 'Quality Check' list and I would double check and triple check the code done there. I would move them to 'Done' when I was satisfied. 'Start Tomorrow' would be reserved for things I want to start the next day, and I always decided on those tasks when I was winding down for the day and wanted a quick 'Up and at Them' list for the next morning. This speeds up efficiency on the level of limiting guess work. This Kanban board used in this way can add a great 'flow' or 'pace' to The process of project completion. Trello is the name of the Web App that I used.

Requirement Execution By Code and Other Highlights



Figures 1 & 2: Protecting sensitive data.

This was a feature that I felt like bringing to light, even though we learned this convention here at Centriq. I have always thought this was a very good thing to add to a project. I created a folder in the UI layer called 'configs'. In that folder houses two XML files 'Connections.config' and 'AppSecretKeys.config'. Through the 'web.config' I direct the Connection Strings to the 'Connections.config' that holds the connection strings to my live data base. We don't want to deploy our database username and password all willy nilly. We need to protect that. In any controller that sends an E-mail it will use what I housed the E-mail client credentials in the 'AppSecretKeys.config'.



Figure 3: E-mail validation Java Script based Captcha (Video game DOOM)

This was such a joy to add, I love the DOOM franchise ever since the first games. When I had built my Portfolio Site, I'd periodically check my E-mail and realize that

after a while I was getting spam. I didn't have any captcha validation on my contact form. This is something I wanted to solve, as a real world client would most likely request. You fill out the contact form and it will not validate unless you play the game and kill 4 enemies. There is a fun cheat code as well where if you type 'idkfa' you obtain the game's infamous 'Super Shotgun'. Thank you to whoever took the time to create this!

```
public ActionResult Index()
{
    //Roadblock highlight 1:
    if (Request.IsAuthenticated && User.IsInRole("Admin") || User.IsInRole("Employee"))
    {
        var ownerAssets = db.OwnerAssets.Include(o => o.UserDetail);
        return View(ownerAssets.ToList());
    }
    else
    {
        var currentUser = User.Identity.GetUserId();
        string userID = User.Identity.GetUserId();
        var userAssets = db.OwnerAssets.Where(a => a.UserID == userID).Include(o => o.UserDetail);

        return View(userAssets.ToList());
    }
}
```

Figure 4: Filtering between roles (GET, Index, OwnerAssets)

This was a bit of filtering to fulfill a project requirement. On the Index GET action in the OwnerAssets controller, roles 'Admin' and 'Employee' should be able to see all assets in the database, where the 'User' level profiles should only be able to see the ones they created. Red: 'Admin' and 'Employee' filter. Blue: 'User' or 'Client' level filtering.

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "ReservationID,OwnerAssetID,LocationID,ReservationDate")] Reservation reservation)
{
    if (ModelState.IsValid)
    {
        int reservationCount = db.Reservations.Where(x => x.ReservationDate == reservation.ReservationDate && x.LocationID == reservation.LocationID).Count();//gets the count where bookings
        //are at the same place and the same date.
        Location locations = db.Locations.Find(reservation.LocationID);//this goes to the db for Locations and finds the value for
        //what the current model's location is. We can access the props from this. Such as the reservatin limit prop.
        if (locations.ReservationLimit > reservationCount)//if the count (which is the bookings at the same date and same place)
        //are less than the daily limit, then it will be created.
        {
            var currentUser = User.Identity.GetUserName();
            db.Reservations.Add(reservation);
            db.SaveChanges();

            string subject = $"[currentUser] added a new reservation";
            string message = $"You have a new reservation! On Date:{reservation.ReservationDate.ToShortDateString()}. At Studio: {reservation.Location.LocationName}. Customer: {currentUser}.";
            MailMessage mm = new MailMessage(
                ConfigurationManager.AppSettings["EmailUser"].ToString(),
                ConfigurationManager.AppSettings["EmailTo"].ToString(),
                subject,
                message
            );

            SmtpClient client = new SmtpClient(ConfigurationManager.AppSettings["EmailClient"].ToString());
            client.Port = 8889; //this is for testing locally.
            client.Credentials = new NetworkCredential(ConfigurationManager.AppSettings["EmailUser"].ToString(), ConfigurationManager.AppSettings["EmailPass"].ToString());

            client.Send(mm);

            MailMessage mm2 = new MailMessage(
                ConfigurationManager.AppSettings["EmailUser"].ToString(),
                currentUser,
                subject,
                message
            );
            client.Send(mm2);
        }
    }
}
```

Figure 5: Reservation Limits and Custom Error View

I needed to fulfill another requirement for this project here. I knew that I needed to utilize the 'reservation limit' field on the 'Locations' table. When creating a location, the 'admin' fills a non nullable field (int). This is intended to be included in a function. If the 'User' level profile tries to reserve time at a location, on a date that has already met it's limit for that day, it returns a custom error page with instructions on how to be considered to fill any cancellations should they happen. I included a contact button on the custom error view so that the flow to the user isn't broken. This

Is a compound feature on my page where it also sends a confirmation E-mail if Create POST was successful. This sends to the 'User' level profile and the 'Admin'.

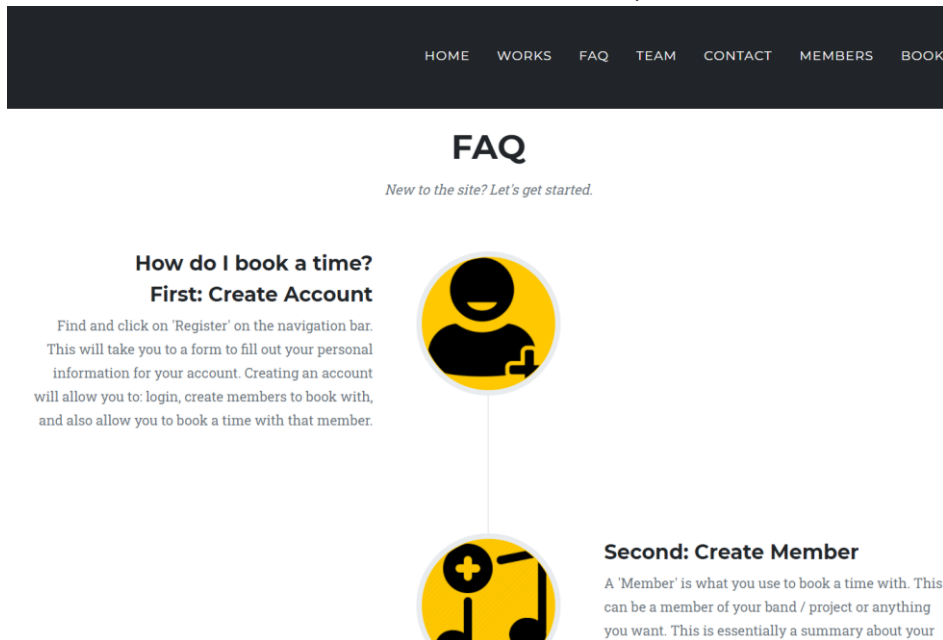


Figure 6: FAQ

I focused heavily on web site conventions, client flow state, and client direction. I wanted to keep this very real world. I need to not only think of my client but also their clients as well. This FAQ was designed to answer simple 'first time user' questions. Any common issue they may experience I tried to reference here and provide a solution.

```
// GET: Reservations/Create
public ActionResult Create()
{
    if (Request.IsAuthenticated && User.IsInRole("Admin") || User.IsInRole("Employee"))
    {
        ViewBag.LocationID = new SelectList(db.Locations, "LocationID", "LocationName");
        ViewBag.OwnerAssetID = new SelectList(db.OwnerAssets, "OwnerAssetID", "AssetName");
    }
    else
    {
        var currentUser = User.Identity.GetUserId();
        ViewBag.LocationID = new SelectList(db.Locations, "LocationID", "LocationName");
        ViewBag.OwnerAssetID = new SelectList(db.OwnerAssets.Where(o => o.UserID == currentUser && o.IsActive), "OwnerAssetID", "AssetName");
    }

    return View();
}
```



While booking a time, why can't I find the member I'd like to book with?

Activate / Reactivate Member

Not to fret! Find and click 'Members' on the navigation bar. Next find and click the member you'd like to book with. Find and click the 'Edit' icon, from here you want to 'Activate' the member by clicking the 'Active Musician?' check box. Click 'Save' and you will now be able to book with this member.

Figure 7 and 8: An example of using the FAQ to solve something

So, focused on real world implementation I needed to not allow the 'User' level profile to book a time with a member that is toggled as 'Inactive'. I needed to ensure bad data wasn't able to be created and stored in the database. See the filtering here.



Jordan Hickman

.NET FULL STACK DEVELOPER

Who I Am

A dedicated, hardworking, self driven team mate who is looking for web development opportunities. I harness a positive attitude and wide skill set in addition to being a Centriq Training graduate. I possess 11 years of work force history primarily as an educator and team leader. Proven ability to establish and maintain excellent communication and relationships with clients. Committed to utilizing my skills to further the mission of a company.

Education

.NET Full Stack Developer, Centriq Training, Kansas City

JANUARY 2021 — JUNE 2021

General Education, Liberty Senior High School, Liberty, MO

Achievements and Recognitions

Student of the Month Award, Centriq Training

JUNE 2021

I was elected student of the month at Centriq Training for displaying my dedication and achievements in the classroom. I was nominated by my instructors and advisors by feedback of my technical skills and knowledge. They had felt that I was very deserving of the award.

Independent Development Projects

Small Business Project (Front End)

A static website for a mock business to promote educational classes of a specific subject. Using strictly custom HTML5 and CSS3 with restrictions on Bootstrap.

Personal Portfolio Site (ASP.NET MVC),

jordanhickman.com/; A self-made site to portray coding achievements in the Full Stack Development field. This site contains portfolio projects and resume and was created using JavaScript, jQuery, HTML5, and CSS3.

Store Front E-Commerce (ASP.NET MVC)

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. Using SQL for databases.

C# Dungeon Game (Console Application)

Created a Windows console application text-based Dungeon game. Using C# fundamentals, I created this by demonstrating logic, polymorphism, inheritance, fully qualified construction of classes, and a passion for early computer games.

Final Project (ASP.NET MVC),

Contact

Kansas City, Missouri

816-820-3938

jdhickman816@outlook.com

Links

[Linked In](#)

JordanHickman.com

Skills

Git

HTML5

CSS3

C# Fundamentals

JavaScript

Bootstrap

jQuery

SQL

ASP.NET MVC

JSON

React JS

Agile Project Management

Created a secure data driven ASP.NET MVC application for 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.

Employment History

Barista, Filling Station Coffee, Kansas City, MO

JUNE 2020 — PRESENT

- Prepare and serve hot and cold beverages, such as coffee, espresso drinks, or teas.
- Describe menu items to customers or suggest products that might appeal to them.
- Receive and process customer payments.

Bartender, The Big Rip Brewing Company, Kansas City, MO

JUNE 2020 — PRESENT

- Hold state and county requirement of liquor and food handler licenses.
- Actively listen to customers to determine beverage preferences and make recommendations.
- Eye for detail in developing new recipes and comprehension of drink mixing techniques.

Production Coffee Roaster, Team Manager, Trainer, Hammerhand Coffee, Liberty, MO

OCTOBER 2016 — JUNE 2020

- Became fluent in roasting coffee orders on a five kilo Mill City roaster.
- Highlighted qualities from coffee roasting to coffee extractions.
- Worked with owner to continuously seek out education for self and store and incorporated that knowledge into the evolving training program.

Front-of-House Director, Team Manager, Trainer, Doughnut Lounge Cocktails and Cafe, Kansas City, MO

OCTOBER 2015 — OCTOBER 2016

- Collaborated with the owner prior to grand opening to establish front-of-house guidelines.
- Created training program for front-of-house, customer service, and coffee service standards.
- Fluent in all shift managing duties, such as, drawer counting, bank deposits, check writing, invoice payments, supervising team duties, and owning store key.
- Created accounts and business relationships with all suppliers for the store.

Lead Barista, Team Manager, Trainer, Kaldi's Coffee, Kansas City, MO

OCTOBER 2013 — OCTOBER 2015

- Scheduled and implemented all store coffee education.
- Conducted pre-employment coffee tasting and knowledge evaluations during hiring process.
- Worked with Retail Store General Manager to continually evaluate and deliver Team Member training adherence and skill development results.
- Monitored espresso drinks, all versions of brew to order coffee, and tea and water quality and maintain adherence to company beverage standards.

