



Coding Bootcamp Survival Kit

Learning Management System – Capstone Project

Danielle Smalley | Full Stack Web Developer | Presentation Date: 8/19/2021



www.daniellethedev.com

smalley.danielle@gmail.com

<https://www.linkedin.com/in/danielle-smalley/>

About Myself + This Project:

I originally went to college to be a teacher, so the Learning Management System prompt was an easy choice. This project allowed me to bring my background in and love for education while showing a bit of my personality at the same time. Prior to and during my time at Centriq, I referenced many outside opinions and experiences when it came to going through a coding bootcamp and being a new junior developer. Because of this, I thought it would be neat to compile some of what I've gathered, along with my own thoughts, into one project that creates a LMS for coding bootcamp students and new grads out in the work force.

Over the past 9 years, I've been client-facing in mainly customer service and sales type roles. At each company, there were continuing education requirements. I remember some of the pain points I experienced as a user, so being on the development side of a continued education system was a blast. Through working on this project, I was able to combine my experience in both working with clients and being a client myself. This is my goal with coding in the future—to always keep the end user in mind.

My favorite part about coding is the ability to create: to create solutions, create art, create something to make life easier for someone, create relationships with fellow developers and clients alike, create a path to overcoming challenges, and more.

Going forward, an area of coding not covered in the track that I'm interested in learning more about would be AI and further work with React. I have a passion for front-end as well as data, so I will be digging deeper into these areas.

Project Synopsis:

- **What:** An ASP.Net MVC, data-driven application with login capabilities and CRUD functionality that is both desktop and mobile friendly. The app requires Identity, PDF & Video upload, contact form, and progress reporting to be implemented.
- **Why:** A learning management system is needed to help manage employees' continuing education requirements. Employees must be able to complete lessons & courses, as well as track their progress. Managers need to be able to monitor progress of their employees, and HR Administrators need to be able to manage active/inactive lessons and courses, as well as users in the system.
- **Who:** Employees, Managers, and HR Admins
- **Where:** Built within Visual Studio and Microsoft SQL Server Management Studio
- **When:** The coding window started on Monday, 8/9/21 with code freeze occurring first thing Wednesday morning, 8/18/21.

Technology Used:

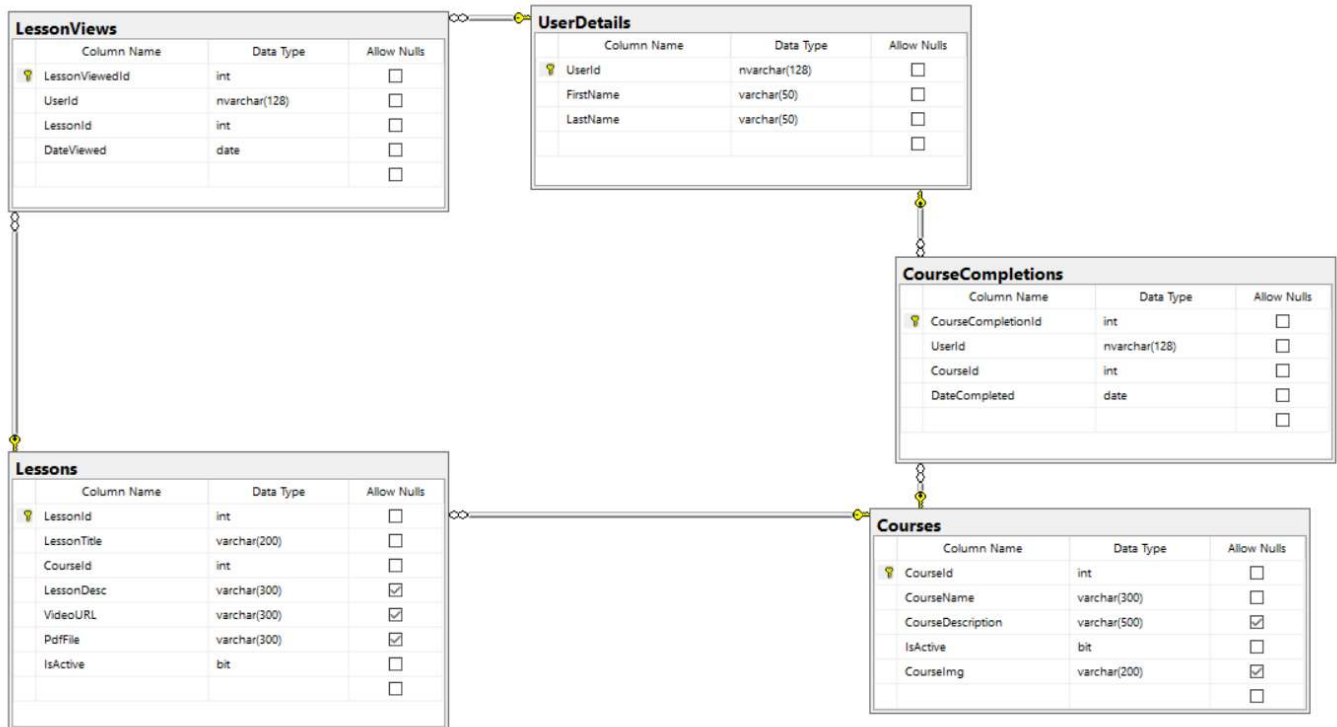
HTML5 | CSS3 | JavaScript | C# | Visual Studio | ASP.Net | MVC | Entity Frameworks | SQL Server Management Studio | Identity Samples | jQuery / jQuery DataTables | Bootstrap | Debugging | CRUD | Trello | Zoom | Excel | FileZilla | SmarterASP

Use/Case Diagram:

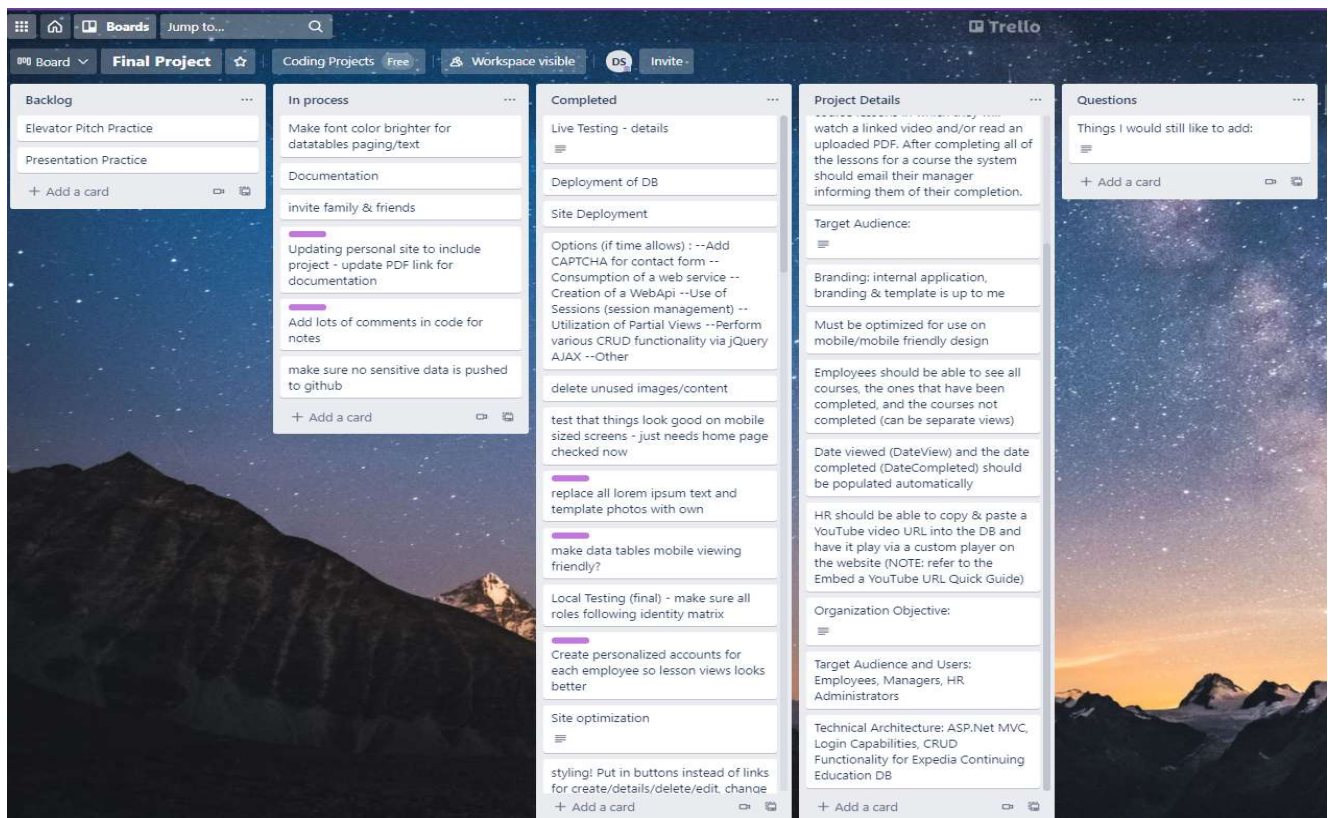
HomeController	Index	Contact	Register	Login			Users & Roles	Index	Details	Create	Edit	Delete			
Employee	X	X	X	X			Employee								
Manager	X	X	X	X			Manager								
Admin	X	X	X	X			Admin	X	X	X	X	X			
Anonymous	X	X	X	X			Anonymous								
Courses	Archived	Active	Detail	Create	Edit	Delete	<p>Ended up adding in:</p> <p>PersonalProgress for CourseCompletions and LessonViews.</p> <p>Employees can only see their own of each. These views only show for employees</p>								
Employee		X	X												
Manager		X	X												
Admin	X	X	X	X	X	X									
Anonymous															
Lessons	Index	Details	Create	Edit	Delete		CourseCompletions	Index	Details	Create	Edit	Delete			
Employee	X	X					Employee	X	X					for themselves only	
Manager	X	X					Manager	X	X						
Admin	X	X	X	X	X		Admin	X	X	X	X	X			
Anonymous							Anonymous								
UserDetails	Index	Details	Create	Edit	Delete		LessonViews	Index	Details	Create	Edit	Delete			
Employee							Employee	X	X					for themselves only	
Manager							Manager	X	X						
Admin	X	X	X	X	X		Admin	X	X	X	X	X			
Anonymous							Anonymous								

- **Anonymous Users:** Able to only see non-data related information. New employees are to Register for their account to begin viewing lessons and completing courses.
- **Employees:** Able to view lessons & courses, and view their lesson & course completion progress. When they complete all lessons in a course, an email is sent to their Manager confirming a course has been completed.
- **Managers:** Able to view employee progress toward completing the 6 courses per year requirement. They will receive an email notification each time an employee completes a course.
- **HR Admin:** Able to add/manage users on the site, have full CRUD functionality for lessons and courses, manage both active and inactive lessons & courses, and upload PDFs or YouTube URLs to each lesson for employees to consume.

Schema:



Trello For Project Management: (end of project view)



Code Snippets:

(Below:) I needed a way to mark a course complete once all lessons were completed so the app will send an email to the Manager when an employee completes a course. The code below encapsulates the logic I built in the LessonsController to handle this:

```
int coursesToLessonCount = db.Lessons.Where(x => x.CourseId == lesson.CourseId && x.IsActive == true).Count();
int coursesCompletedCount = db.LessonViews.Where(x => x.Lesson.CourseId == lesson.CourseId && x.UserId == userid && x.Lesson.IsActive == true).Count();

if (User.IsInRole("Employee") && coursesToLessonCount == coursesCompletedCount)
{
    CourseCompletion completion = new CourseCompletion();
    completion.UserId = userid;
    completion.CourseId = lesson.CourseId;
    completion.DateCompleted = DateTime.Now;

    var firstCompletion = db.CourseCompletions.Where(x => x.UserId == userid && x.CourseId == lesson.CourseId).FirstOrDefault();
    if (firstCompletion == null)
    {
        db.CourseCompletions.Add(completion);
        db.SaveChanges();

        string courseTaker = db.UserDetails.Where(x => x.UserId == userid).FirstOrDefault().FullName;
        string completedCourse = db.Courses.Where(x => x.CourseId == lesson.CourseId).FirstOrDefault().CourseName;
        var completionDate = completion.DateCompleted;

        string courseCompletedMessage = $"{completedCourse} course was completed at {completionDate:g} by {courseTaker}.";

        MailMessage mmsg = new MailMessage(ConfigurationManager.AppSettings["EmailUser"].ToString(), ConfigurationManager.AppSettings["EmailTo"].ToString(), "Course Completed",
            courseCompletedMessage);

        mmsg.IsBodyHtml = true;
        mmsg.Priority = MailPriority.High;

        SmtpClient client = new SmtpClient(ConfigurationManager.AppSettings["EmailClient"].ToString());

        client.Credentials = new NetworkCredential(ConfigurationManager.AppSettings["EmailUser"].ToString(), ConfigurationManager.AppSettings["EmailPass"].ToString());

        try
        {
            client.Send(mmsg);
        }
        catch (Exception ex)
        {
            ViewBag.ErrorMessage = $"Oh heck, the course completion notification email could not successfully send to your manager. Please try again later. Error Message: <br>";
        }
    }
}
```

```
// GET: Courses
[Authorize(Roles = "Admin, HRAdmin, Manager, Employee")]
public ActionResult Index()
{
    //want to compare courses vs completed courses for employees
    string userid = User.Identity.GetUserId();
    var completedCourses = db.CourseCompletions.Where(x => x.UserId == userid).ToList();

    var activeCourses = db.Courses.Where(x => x.IsActive == true).ToList();
    foreach (var item in completedCourses)
    {
        foreach (var aCourse in activeCourses)
        {
            if (aCourse.CourseId == item.CourseId)
            {
                aCourse.HasCompleted = true;
            }
            //if we found a match here, this is completed
            //otherwise, = false
        }
    }

    return View(activeCourses.ToList());
}

[Authorize(Roles = "Admin, HRAdmin")]
public ActionResult IndexInactiveOnly()
{
    var inactiveCourses = db.Courses.Where(x => x.IsActive == false).ToList();
    return View(inactiveCourses);
}
```

(Left:) Since HR Admins need to be able to manage both active and inactive courses, I built this logic in the CoursesController so admins can toggle back and forth between active vs inactive. Similar logic is built in the LessonsController to toggle between active & inactive lessons.

Code Snippets Continued:

(Below:) I incorporated CAPTCHA/RECAPTCHA from Google's API in my contact form to help prevent spammy messages (this had been a problem previously on my personal site). This is within the Controller, a few additional lines are needed in the web.config but this is the main chunk of code:

```
[HttpPost]
[ValidateAntiForgeryToken]
[CaptchaValidator(PublicKey = "XXXXXXXXXXXXXXXXXXXX", PrivateKey = "XXXXXXXXXXXXXXXXXXXX", ErrorMessage = "Invalid input CAPTCHA", RequiredMessage = "Please verify you are not a robot (beep boop)")]
public ActionResult Contact(ContactViewModel cvm, bool captchaValid)
{
    if (!ModelState.IsValid)
    {
        return View(cvm);
    }

    if (ModelState.IsValid && captchaValid)
    {
        string message = $"You have received an email from {cvm.Name} with a subject of {cvm.Subject}. Please respond to {cvm.Email} with your response to the following message: <br/>{cvm.Message}";

        MailMessage mm = new MailMessage(ConfigurationManager.AppSettings["EmailUser"].ToString(), ConfigurationManager.AppSettings["EmailTo"].ToString(), cvm.Subject, message);

        mm.IsBodyHtml = true;
        mm.Priority = MailPriority.High;
        mm.ReplyToList.Add(cvm.Email);

        SmtpClient client = new SmtpClient(ConfigurationManager.AppSettings["EmailClient"].ToString());
        //client.Port = 8889; - for gmail

        client.Credentials = new NetworkCredential(ConfigurationManager.AppSettings["EmailUser"].ToString(), ConfigurationManager.AppSettings["EmailPass"].ToString());

        try
        {
            client.Send(mm);
        }
        catch (Exception ex)
        {
            ViewBag.CustomerMessage = $"Ope. Something went wrong and we couldn't process your request. Please try again later. Error Message: <br/>{ex.StackTrace}";
            return View(cvm);
        }

        return View("EmailConfirmation", cvm);
    }
}
```

(Below:) Each lesson needed to contain either a PDF or YouTube URL for the employee to view to complete the lesson. Within the Details view for Lessons, I built code to house these items:

```
<dt>
    @Html.DisplayNameFor(model => model.VideoURL)
</dt>
<dd>
    @if (Model.VideoURL == null)
    {
        <div>
            <p>No video for this lesson; please just view PDF</p>
        </div>
    }
    else
    {
        <iframe id="ytplayer" type="text/html" width="640" height="360" src="@Model.VideoURL" frameborder="0" title="@Model.LessonTitle lesson video"></iframe>
    }
</dd>
<dt>
    @Html.DisplayNameFor(model => model.PdfFile)
</dt>
<dd>
    <a class="btn btn-info" href="@Url.Content("~/Content/img/pdf/" + @Model.PdfFile)" target="_blank">Click For Lesson PDF</a>
</dd>
</div>
</div>
</div>

if (lesson.VideoURL != null)
{
    if (lesson.VideoURL.Contains("/watch"))
    {
        lesson.VideoURL = lesson.VideoURL.Replace("/watch?v=", "/embed/");
    }
}
```

(Left:) Code added to the LessonsController – Create for incorporating embedded YouTube videos.

DANIELLE SMALLEY

573-818-5658

SMALLEY.DANIELLE@GMAIL.COM

WWW.DANIELLETHEDEV.COM

[LINKEDIN](#)

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/Analytical Thinking, Communication, Project Management, Collaboration, Strategic Planning, Time Management, Problem Solving, Detail-Oriented, team player with excellent verbal and written communication skills.

INDEPENDENT DEVELOPMENT PROJECTS

- **Personal Site:** www.daniellethedevelop.com
 - **StoreFront:** Created a secure application for managing product data. Application is built to simulate an online store front with a session based shopping cart. Administrators can manage product, category, and employee data.
 - **Pair Programming – 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.
 - **Capstone Project – Learning Management System:** Created a secure, data-driven ASP.NET MVC application from design through deployment for managing continued education within a company. Administrators can manage required courses, lessons, employee progress, and user details.
-

TECHNICAL TRAINING AND ADDITIONAL EDUCATION

Full Stack Web Developer Program, Centriq Training
Kansas City, MO (Virtual)

Graduated 08/20/2021

Featured Student of the Month – August 2021

- MVC Framework
- Trouble Shooting & Debugging
- Source Control
- Agile/Scrum (Created Team Project)

- Website Design & Deployment
- Pair Programming
- Code Review
- Object Oriented Programming

Ozarks Technical Community College
Lebanon, MO

08/2011 – 05/2013

- Associates of Arts and Teaching

WORK EXPERIENCE

Home Buying Concierge, Veterans United Home Loans
Columbia, MO

09/2019 – 02/2021

- Connected with customers at various points of the home buying process. Acted as the customer's advocate, advisor, and project manager.
- Collaborated with internal and external providers on obtaining products/services as needed.
- Contributed to new third party service ideas to expand business offerings, new scripting ideas for training, and improvements to procedures to increase efficiency and speed of service.
- Tracked procedural improvements and changes via Tableau and Excel.
- Trained/mentored new employees.

Assistant Loan Officer/Transaction Coordinator,
Veterans United Home Loans
Columbia, MO

01/2016 – 09/2019

- Effectively communicated with borrowers, loan officers, underwriters, closers, and other 3rd parties involved in the transaction; responded promptly to inquiries and requests via phone, email, and/or text/IM.
- Effectively managed a pipeline count of 30-50 borrowers each month from loan application to funding within required turn times and quality standards.
- Requested, received, reviewed, and organized all necessary items to complete loan approval.
- Reviewed and analyzed credit, income, and asset documentation; established conditions with company lender guidelines and maintained ownership of the file quality.
- Attended training to create progression in the role and participated in numerous pilot programs to test new client-facing technology or updates to internal systems/programs used.

Service Center Support, FedEx Freight
Columbia, MO

10/2013 – 12/2015

- Billed all invoices for incoming and outgoing freight (data-entry heavy).
- Checked-in driver's and reviewed all driver/freight paperwork.
- Data entry, filing, and record maintenance.
- Answered incoming calls to freight center, assisted clients and other employees with freight inquiries.

Customer Service Representative II, Central Bank of
Boone County
Columbia, MO

05/2013 – 12/2015

- Promptly responded to incoming calls, emails, and chats from customers needing assistance in a variety of areas.
- Performed account maintenance and solution tracking via CRM.
- Fulfilled customer service functions.
- Completed special projects as assigned.

911 Operator/Dispatcher, Laclede County
Sheriff's Office
Lebanon, MO

05/2012 – 05/2013

- Answered incoming 911 calls and dispatched police, fire, and ambulance accordingly.
- Answered incoming non-emergency line calls and escalated/resolved accordingly.
- Performed other duties as assigned by county offices.