Web Application With Attached Database

Trevor A. Engstrom

Saint Martin’s University

# Abstract

I built a web application with an attached SQL Server database for my senior project. The web application was built primarily in the C# language with Visual Studio and SQL Server Management Studio. The web application provides a web site for visitors to browse the inventory of a truck dealer while having a dashboard for employees to update the inventory and upload photos for the website.

*Keywords:* web application, web development, SQL, EF6, database, Visual Studio

# 

# 

# Web Application With SQL Server Database

For my project I wanted to build a web application. When I started the project in January, I was just learning basic web development skills but I really enjoyed what I was learning and was eager to get into more advanced topics. I found a client in Fort Myers Florida who wanted a website for his new business but he didn’t know where to start or how to update the website. I told him I could help him by creating a product that would meet his needs and also drive business to his new venture.

# Method

## Literature Review

There are many systems on the market that are similar to the one I designed. A web application with an attached database and functioning database is not a new concept but it is an idea that is constantly being refined. There are many off the shelf products that let a user pay either a large fee up front or a monthly fee to build their own website or online store. The issue with these is that they are becoming so popular that it is easy to spot one of these templates. For me, when I see an online marketplace or vendor’s website that uses one of these off-the-shelf templates or easy to self design websites, they tend to lose credibility with me because I know they took a shortcut to get there. Even though the design is supposed to be unique, they all tend to have a similar look and feel to them.

## Design Methodology and Design

The basic design methodology for this project were to design something streamlined and simple to use for the end user. That methodology drove the project to end up using a web application, with a simple database, and an easy to use dashboard.

**Requirements.** Functional requirements for this project included the ability for the web application to communicate with an instance of SQL Server database. The user needed a dashboard with an easy-to-use database that included ways to perform CRUD operations on the database as well as a way to handle file uploads.

**Use Cases.** There are two main users for use cases; the owner and website visitors. The owner will be using the app to log in and update their inventory that is displayed on the website. The other use case is visitors to the website and potential customers interacting with the website which hopefully leads to them contacting the business and eventually purchasing a vehicle.

**Physical System.** There is no real physical system for this project. The project was built with the understanding that the client would have no physical resources besides accessing the project through an internet browser from their own network. The app will all be hosted from the cloud.

**User Interface.** As with use cases, there are basically two seperate user interfaces. There is the portion of the website the visitor can visit which includes all pages except the login, account information, and the dashboard. The visitor’s experience includes a homepage that gives a rundown of the site and links to the other, more informative, portions of the website. There is also a navigation bar in the header that can bring the user anywhere they have access. The visitor can view the inventory page which includes a datatable with all the summary information for the dealership’s inventory as well as a small photo of each vehicle. The user can click anywhere on the row and be taken to a details page that shows additional information about each truck as well as a photo gallery for the truck.

The user interface for the employees includes every the visitor can see, plus the login, account, and dashboard pages. The dashboard page includes a datatable with the dealership’s inventory. The employee can perform CRUD operations on the table which will immediately be visible to the visitor. At the bottom of the dashboard, employees can upload photos and assign them to a particular vehicle in the inventory.

**Database and System Interaction.** The database is an instance of Microsoft SQL Server. The database only has two tables, ‘Inventory’ and ‘Photos’. The Photos table has a foreign key to the primary key of the Inventory table. The photos table stores a file path to where the actual photo is stored.

I used Entity Framework 6 as an object-relational model to interact with the database with C# from within Visual Studio. Entity Framework Core is the latest version of the Entity Framework series but it had some features that I was uncomfortable with, such as the lack of EDMX files that allow you to change your database with a diagram through Visual Studio. Entity Framework Core does all database updates and syncing through a command line migration tool. I felt more comfortable with EF6, and was still able to access the database securely, so everything worked out in the end.

## Implementation

I was a team of one so I did not have any coordination or planning to conduct with others, but I still developed a plan and tried to stick to it while making adjustments along the way.

**Development Methods.**

I’ve been learning about the Agile development method and tried to implement a similar system for myself for the project. I started with a defined goal and established a basic timeline I wanted to follow. I knew I needed the application done by March but my goal was to have it fully operational, tested, and deployed by the end of December. Due to improper planning, I failed to meet my original objective. At my job, we work in two week sprints, where we set goals for the two weeks and outline what tasks we plan to finish and estimate how much time and effort those tasks will take. I tried doing a similar, although less formal and less documented system with this project.

**Completion Process.**

In an effort to complete this project on time, I tried to stick to my timeline as much as I could. I understand the repercussions of missing deadlines and planning to make up for the time on the back end so that was something I wanted to avoid. For this project, missing deadlines was unavoidable. I set too strict of a timeline and tried to do too much into the time I gave myself.

I wanted to focus on functionality first before I worried about the look and feel of the app. I knew that if things were functional, then I would have less stress about the project and be in a better place to properly design the application. Although I was unable to complete the project by the initial deadline I set for myself, I was able to get the project operational and working.

## Experimental Result and Analysis

Because the project is still unfinished, I’ve been unable to test the web application at full load in a deployed environment. The main requirement from my client, that he repeated over and over was that he wanted something easy to use because his level of computer literacy maxes out at basic excel operations. To conduct an impromptu test, I used the most computer illiterate person I know as a test subject, my wife. I deployed the website to a local host and told her to look at what trucks were for sale. She was able to easily get to the inventory page. I then showed her the dashboard page and told her to add her car to the database. She easily found the “add” button and followed the prompts to add her car to the database. I then told her to upload some pictures that would represent her car. She used the drag and drop method to drop two pictures into the uploader. I had to explain to her what to write in for the inventory ID textbox but that will eventually be a dropdown box and be almost impossible to miss. The unstructured test was successful. The client wanted something easy to use and I proved that I built an easy to use system by having a person with no prior knowledge of the system or car dealerships update the inventory and upload pictures to be displayed on the website.

## Conclusion and Future Work

This project was hard, time intensive, and probably too much for a beginner to handle by themselves but it was an extremely valuable experience. This project allowed me to gain some valuable experience and ultimately helped me obtain a job creating web applications. I plan to continue working on this project in preparation to have it go live sometime in March 2020. This will be my second website I’ve ever put online but my first real major website. It will be good to have something out in the world where I can point people to see what I am capable of.