Razor Sharp

UniversityBarberShop.com

Kevin Nguyen, Tanner Livingston, Anderson Porta

12-6-2019

CS 4540 - Final Deliverables Report

ABSTRACT

UniversityBarberShop.com is a website useful for people looking to schedule appointments with barbers, hair stylists, etc. On the website, learn about the workers, make appointments with them, read reviews by other people, and make comments and reviews of barbers. This website has application in the real world in that barbershops could implement this website and have customers make appointments, read reviews, and overall learn about the barbershop. While we made this website specific to fictional barber shop, a UI change could make thiec2-18-232-121-124.compute-1.amazonaws.coms a Nail salon, or a day spa.

URLs

Kevin Nguyen: ec2-34-226-208-84.compute-1.amazonaws.com

Tanner Livingston: ec2-18-232-121-124.compute-1.amazonaws.com

Anderson Porta: ec2-18-212-29-127.compute-1.amazonaws.com

INTRODUCTION

The project is aimed to help small time barber shops, nail salons, or small companies looking to advertise and schedule services. We made this website specifically for barbers, but a UI change could make it useful for nail technicians, and so on. The general purpose of the website is to be able to read about the workers, make appointments with workers, read reviews and make reviews of workers.

When you enter the website, you'll be greeted with the store's name. Below, you'll see a list of barbers at this specific shop with their name, a quick view of when services they provide and their profile. Upon clicking on a profile, you'll see more about these workers. In the profile page, you'll see their name, the worker's jobs, the services they provide, and a description of themselves. Below are three buttons to choose from.

Upon clicking the book now button, you'll be prompted with 30 minute time slots between 9am to 4:30pm. A customer will be able to book an appointment, and Barbers should be able to see all the appointments as well.

If you go back to the workers page, you can view reviews of a specific barber. These reviews list their name, the rating out of 5 stars that they got, their comment, and when it was posted. These comments can be created, edited or deleted by the Admin themselves, or customers of the shop. When a customer decides to edit or delete their review, their Username must match the comment of the original commenter. This will prompt a Edit or Delete button for those specific comments only.

When deciding to write a comment, you must be logged in, and must be in the Admin or Customer role; this avoids Barbers writing comments to themselves. When writing a comment, you cannot change your name, or the worker. Your name is tied to your username and email, and is uneditable by anyone. If you'd like to change workers, you must navigate to their profile, and click write a comment from there. Admins have the ability to make comments of any kind, with no restrictions, through the admin page.

The admin page consists of the role change page, as well as links to the pages to 'CRUD' Workers and WorkerComments.

FEATURE TABLE

Feature Name	Scope	Primary Programmer	Time Spent	File/Function	Lines of Code
Identity Login and Roles, role change ability, Permissions	Back-end	Kevin Nguyen	4 hours	Admin.js, Admin.cshtml, AdminController.cs, DBInitializer.cs	342
CRUD Worker Comments	Back-end, Front-end	Kevin Nguyen	8 hours	WorkerCommentsController.cs, DBInitializer.cs, Some /WorkerComments/ view files	300
Customer Edit/Delete their own comments	Back-end	Kevin Nguyen	5 hours	WorkerCommentsController.cs	150
Star Ratings for comments	Front end	Kevin Nguyen	2 hour	/WorkerComments/Details.cshtml, ShowComments, Index,	50
Barber functionality	Back-end	Tanner Livingston	8 hours	WorkersController.cs, book.js	200
Barber Profile/Model	Back-end	Tanner Livingston	6 hours	Worker.cs, DaySchedule.cs, Some in DBInitializer	150
Book appointment	Front-end	Tanner Livingston	5 hours	Index.cshtml, Book.cshtml	230
Landing Page, Navigation Bar, and Footer	UI/Front-end	Anderson Porta	7 hours	View/Home/Index.cshtml Landing-page.css, site.css, site.js	323
Login and Register Forms	UI/Front end	Anderson Porta	3 hours	Login.cshtml, Register.cshtml, login.css	176
Barber Profile Page	UI/Front-end	Anderson Porta	1 hours	View/Workers/Details.cshtml details.css	108

Edit Barber Profile, View Comments, Write Comment	UI/Front-end	Anderson Porta	2 hours	View/Worker/Edit.cshtml View/WorkerComments/Create.cshtml and Details.cshtml, comment.css	250
Booking Page	UI/Front-End	Anderson Porta	3 hours	View/Workers/Book.cshtml Booking.css	240
Admin functionalities (Manage Worker, Manage Worker Comments, Role change)	UI/Front-end	Anderson Porta	3 hours	View/Admin/Admin.cshtml table.css	120

INDIVIDUAL CONTRIBUTION

Team member	Time spent	Lines of Code Committed
Kevin Nguyen	19 hours	1,548
Tanner Livingston	19 hours	580
Anderson Porta	19 hours	1,217

Kevin's Contributions:

I was in charge of setting up user login and role management, and customer reviews. Some of my user login and role management code was from the LOT. When adding users and roles, I spent most of my time trying to figure out how to automatically add new registered users to a role upon registering. This consumed a lot of time for me, as it was a necessity. I was also spending time in another branch trying to see if removing the customer role was a good choice, but scrapped the idea. After finishing that functionality, I went to work on comment functionality. This pertained to comments being specific to workers, Admins crud functionality, as well as customers being able to make comments, then edit and delete their own comments. This was where most of my

time was spent, with many hours spent on trying to initialize the database, to trying to figure out how customers can edit/delete their own comments.

Tanner's Contributions:

I was responsible for the data model for Workers, and their database integration. I spent a good amount of time thinking about and planning for different functionalities, especially with regards to the worker's schedule. I spent a while struggling with integrating workers with the users database. The biggest problem with workers was deciding how their schedules would be saved. After many attempts, I ultimately settled on the DaySchedule model that is in the final version. Finally, I managed to get the workers data model, and its related DaySchedule model, migrated and integrated with the database.

I was also responsible for bringing workers and their schedules to the front-end. This meant I needed to scaffold and modify cshtml pages from our data models, and write back-end controller code to query the database in order to retrieve the relevant information for each worker page. Additionally, I added functionality for website users to be able to interact with the schedule. This involved making a dedicated workers controller, with functionality to return a worker's schedule for the next 7 days (creating those days in the database if they didn't already exist), a book.js file to handle users booking appointments, and making a Book.cshtml view that would show the worker's schedule, as well as provide to the controller any relevant information when a user books an appointment.

Anderson's Contributions:

I was in charge of taking care of the User Interface for the whole website. I started researching popular UI designs among websites that are specifically for barbershops. After several hours of research I came up with ideas on how our website should look.

The first page I focused on was the landing page. It took me around seven hours to complete it. Then, I worked on the barbers profile UI, and the Login and Register forms. Once I implemented those pages I felt that I was doing a good job because I was happy with how our website was looking. In addition, I improved the UIs for the Admin page

functionalities and the CRUD pages so they can have a consistent look with the other pages. I think that the hardest page that I worked on was the booking page. The reason is that I tried different types of designs, but none of them looked good. After thinking on how this page should look, I came up with one that I think looks decent.

SUMMARY

We set out to make a site that was simple to use, and that provided the basic functionalities that a worker or user of the site would expect. The level that our team performed was superior. Each team member completed their required work on time. We overcame lots of obstacles both individually, and together as a team. The first and main obstacle we faced was having an understanding of how the site would look and function. When we did our initial planning, we were relatively thorough, yet there were often small details that would come up that we needed to resolve together so we each could effectively carry out our individual tasks. This wasn't a huge issue, though, as we divided the tasks at the beginning to allow for each of us to work largely independently of each other.

We are proud of the work we accomplished, and our site demonstrates some element of almost everything we have covered in CS4540. Our work is a good example of a modern, functional shop-type site. It is at the very least, a functional prototype, and with a little more work, could easily become a fully operational booking website for a Barber/hair/cosmetics shop. In fact, as a result of our design and engineering decisions, it could be easily modified to represent any other kind of shop where customers book appointments and leave reviews.