CSC 337 Group Project

"Guitar World"

An E-Commerce Website

May 4th, 2025

University of Arizona

Professor Reyan Ahmed

Prepared By:

Preston Greenwood

Sai Josiuna

Van Edmundson

Motivation

Our motivation for designing a webpage based on guitars stemmed from several class examples we had learned and studied from, as well as the flexibility and creativity we could showcase. Initially, when discussing ideas to pursue, we wanted a project that we could make unique to us and would have many different avenues of success. A guitar website was something all three of us felt passionate about as we discussed the routes we could take with this project. Basically, we conceptualized a guitar website as a solid foundation that we could start building and adapt our modules as we went. This allowed us to fulfill our project requirements, but most importantly, maintain and present our creativity while coding. Creativity was a core value each of us required when brainstorming ideas for our project.

Module Overview

The User Module manages all of the accounts created on our website, to include registration, login, and session maintenance. When on the website, users can create either a customer or admin account. With this account, they can log in while also maintaining an active session, no matter what page on the website they visit. Login credentials are validated through backend Express routes and logins trigger a server side session. This module is used to let only authenticated users access the changing of account details or viewing of guitars. If a user attempts to access a page restricted to them, they are redirected to the login screen. This maintains our site security. The Store Module is used to showcase all guitars ready to be purchased by the users. You can browse guitars and add them to cart for purchase. Every individual product has their own page with different pictures, specifications and descriptions. These products have their own html page, each page being linked to a specific guitar. When logged in, users can interact with the products page as well as add a guitar they like to their online shopping cart. This cart is managed in the browser using JavaScript and localStorage. Additionally, it integrates well with our project, because in order to purchase a guitar from the store, one has to be authenticated using our User Module. The Account Management Module allows each logged in user to edit their personal information. This includes the individual's first and last name, email, and full shipping address. A user can view their past orders and manage their shopping cart. Past orders are retrieved from the MongoDB database and passed to the

current active user session, ensuring only the users can view their order history. It relies on the User Module to make sure each individual can only view their own information and order history. Altogether, these modules work cleanly and seamlessly, to create a cohesive experience for the individual on our website.

Functionalities

Our guitar website supports many key features that create an interactive experience for the individual shopper. First, users can register an account and log in securely at any time, with the correct username and password. They only need to log in once per session, as the login is maintained across all of our pages. This way, a user can browse not just guitars, but our website as a whole, freely using the navigation bar to explore any page they desire. Secondly, users can browse our full catalog of guitars, each one having a section for a description, its specifications, and additional images. Even if the user is not knowledgeable about a guitar, they can learn more about their individual uniqueness, and easily switch guitar pages to learn more about them. Thirdly, logged in users can view their account information such as past orders, as well as their email, username, and shipping address. This allows the user to conduct their own experience, as well as track their own account with our guitar company. Fourthly, our website features an engaging homepage with customer reviews and a “Visit Us” section provided out of the way, at the bottom. This includes the contact information for our business, so customers can reach out to us with questions or comments. One of the key features we wanted to add, to promote the feel of a real e-commerce store, is the shopping cart. Users can add items to a cart, see their total price, and edit the contents to buy only what they want before checkout. When checking out, cart data is submitted to the server, finalizing the order. Then, users can see a confirmation page. Admins can view all the pending orders and fulfilled orders. Our store becomes fully available when a user logs in (with only some pages restricted to admins). When an individual tries to use a restricted page, it redirects them to the login screen. This means certain pages are only accessible to users. Lastly, all pages are styled with the same revolving gray color gradient, as well as a unique popping color scheme, promoting a cohesiveness to our website and a modern feel.

Technical Details

Our project is built using a modern full stack JavaScript environment, with Node.js and Express handling the server side logic, requests routing, and middleware operations. For important data storage, we used MongoDB, connected using Mongoose, which allows us to keep and interact with the database in a clean and efficient way. We defined Mongoose schemas for user accounts and records, to maximize database efficiency. MongoDB stores all user accounts, orders, and cart data. Passwords are securely hashed before being stored, ensuring proper protection of accounts. This project uses these three modules in unique ways. The User Module handles account creation, login authentication, and session tracking. The Store Module displays guitar listings and manages interactions with individual product pages. And the Account Management Module allows our users to edit their personal profiles, place orders, and view their order history. Session management is done safely through Express using the express session middleware to maintain the login state on our server. Also, the front end uses localStorage to display the login username and handle user specific navigation across pages. This keeps our website easier to use. Each module is in response to specific endpoints, supporting our well organized design that we made sure to prioritize. We built our interface using HTML, CSS, and vanilla JavaScript. We focused on dynamic interactivity, provided by a script file source.js. This file manages some of our key features discussed earlier. Each product page is manually structured with unique images of the guitar the client selects, and the cart totals are calculated and passed to the backend during checkout. Creating files this way promotes a way for customers to shop easily and freely. Our Express server handles both GET and POST requests for page navigation, login, and form submissions. Individuals with admin accounts have additional features, including the ability to view all orders and view fulfilled orders as well as pending ones. To run the site, MongoDB must be installed and running locally, and the correct database URI must be specified on line 7 of server.js. The server runs on localhost:8080, as noted in our README. This setup ensures our application handles both client side interactivity and server side logic in a cohesive and scalable way. Together, the modules, functionalities, and technical details allowed us to fully express our creativity and technical coding skills, resulting in a comprehensive and visually appealing website.