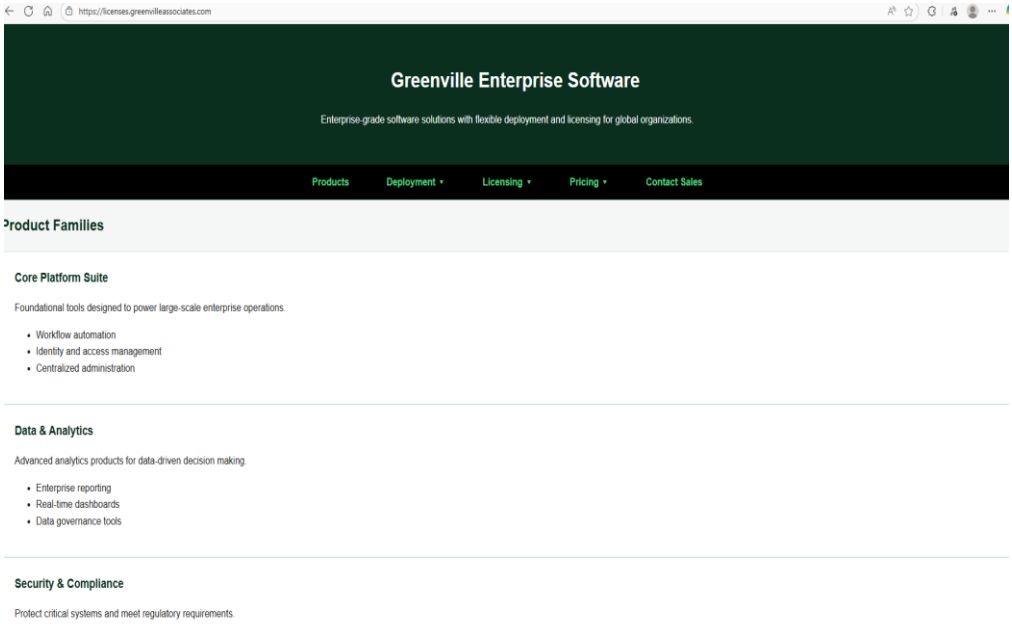
FINAL

WIREFRAMES & SCREEN LAYOUTS & DB ARCH

FOR GREENVILLE LICENSE MANAGER (GEM.LM)

John S. Stritzinger
Computer Science 242 – Wireframe/Page Layout Requirements
Spring 2026

Contents



SECTION 1. EXECUTIVE OVERVIEW

1.1 Definitions of Wireframes

Wireframes are Screen Layouts spanning an entire application. Although there are multiple ways to do screen layouts for Analysis work, this class has discussed using Figma Design and Make Tools with a combination of HTML and React Pages. That was the requirement for this assignment.

1.2 Figma Design

Figma design is similar to other tools from Microsoft and Adobe software and allows a user to created a layer Visual on a canvas. Unlike some tools however Figma has "SnapTo" tools which are built based on a Rendered Device output allowing for developers to make tools for a single type of output.

1.3 Figma Make

Figma Make takes a "Figma Design" or any other HTML file, or Graphic that a LLM AI Tool can ingest and builds HTML, React, and other developer finished tools which can be compiled into real programs. Figma has partnered with ClaudeAI which is Anthropoics AI tool.

1.4 Initial Wireframes Based on GitHub Project Management Theme for 1st half of the course.

We thought that the instructor had intended Wireframes just to be a demonstration of AI tools which we would later have to copy for our final project, but that initially it would be an extension of the website we have built to show our project states/project management stages. As such our initial submissions and our discussion in Section2 were based on this work product which was also completed as designed. The instructor asked us to rework our presentation before final.

1.5 Greenville License Manager as part of Greenville Enterprise Manager

We believe the Greenville License Manager is a sub-component of the Greenville Enterprise Manager Tool and will allow the real-time monitoring of Application Use across an Enterprise which is the first of its kind, using location processing on a per station basis, per region, and hosting shard which are definitions beyond the scope of this document.

SECTION 2. INITIAL WIREFRAME LAYOUTS -> GIT.IO THEME

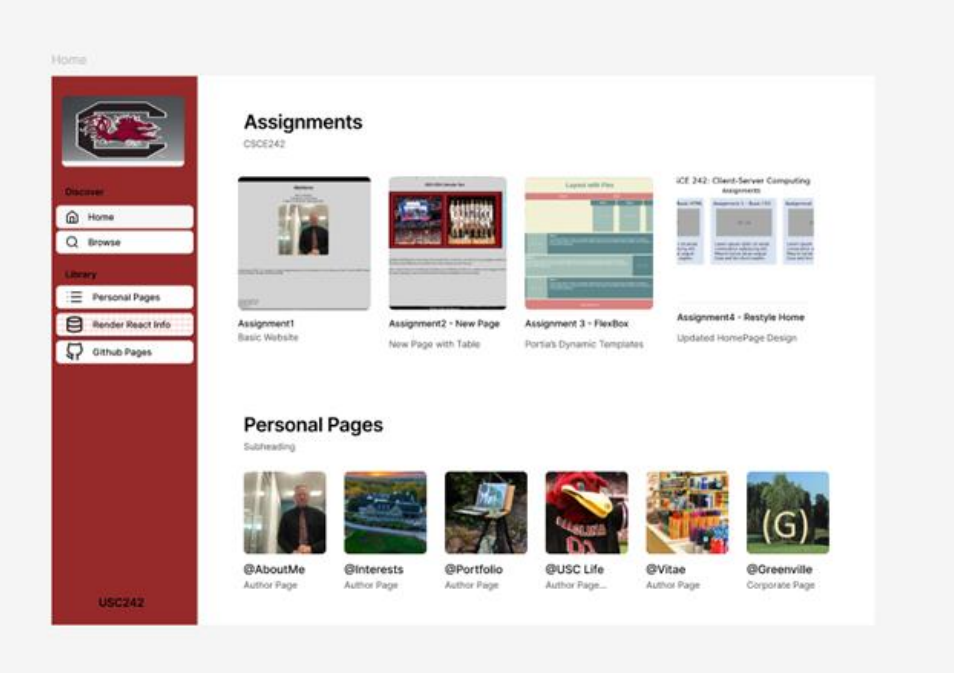


Figure 2.0.1

2.1 Scope of Work

Our initial plan for our wireframes was to build a React improvement for Project Management Professionals which would be a generic form of our class assignment.

GITHUB.IO – IMPROVEMENTS FOR PROJECT MANAGEMENT PROFESSIONALS

Our instructor told us she didn't believe that was consistent with our project after we had already effectively completed the work.

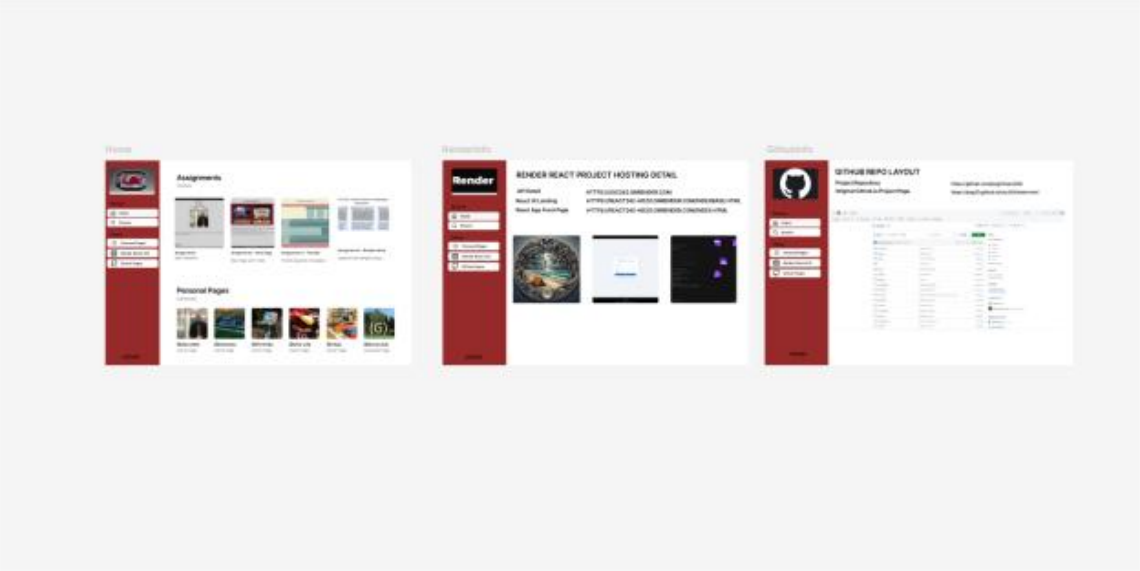
2.2 Scope Confusion

We believed that the Project plan we submitted for our project was going to be our End of Term Solution, and not related to the wireframe solution. As such our initial presentations from Figma were for the Github restyling project. We therefore were prepared to rebuild the entire project management site for our class based on the structure of the project on Github.io which does not support Javascript. We believed we could build a new version which would be a generic project tool which would be database driven with java script enabled on Render as the first part of our project. The instructor has said she didn't intend that to be part of the scope, but I believe its valuable anyway as I am a PMP and I think it would help me for other Consulting Deliverables. We have added two tables for Scopes, and Projects which are included in our APIs on Mongo DB which are working and in production.

2.2 Current Github Site

The current Github site has approximately 7 personal pages with a a vitae, along with an Assignments Section, and Projects Section. The goal of this assignment was to have a unifying flex design for a simple page environment which would be mobile friendly, while the tasks themselves were in separate Github folders. The result is approximately what is available with a Sharepoint Site just with a bit better change control. The instructor using Git can see exactly whats been changed, and when which is an improvement over Sharepoint. Furthermore older versions can be restored.

2.3 Layout For The site



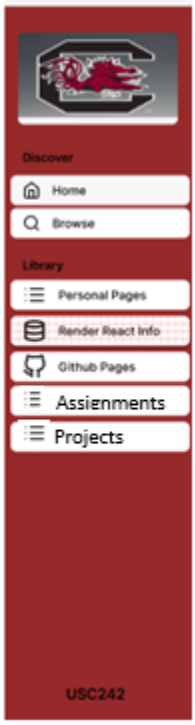
Our initial Wireframe as of above had:

a) Home Page with Links to Personal Latest Assignments, and Project Stages just like are the requirements for the existing page at jssg33.github.io.
b) A Side Navigation bar which allowed someone to switch from the Home Page to Personal Pages, to Assignment Pages, and Product Pages with a Sub-Home Page for Each subsection which is an improvement over the current design we have done in class.
c) This effectively gives us 4 Main Navigation Pages (Home, Personal, Assignments, Projects), all of which we modeled in Figma Design and Two Infrastructure Pages (React, Github).
d) We then gave Figma Make the Image of our home page, and gave it some guidance on the side nav pages, and it built out a React Structure with the 4 main navigation pages, and the approximately 30 sub pages which could only be reached from the main 4… in approximately 10 pages per sub page, **plus the home page which is a highlighter for the current task underway. This would allow us to move old tasks off the homepage when**

they are completed. This could be done by the database by only presenting on the dashboard tasks which are set to "underway" or not complete, and ordered by those sub-tasks.

e) **The Completed First Pass of the Figma Design Board**

2.4 Layout of the Side Panel



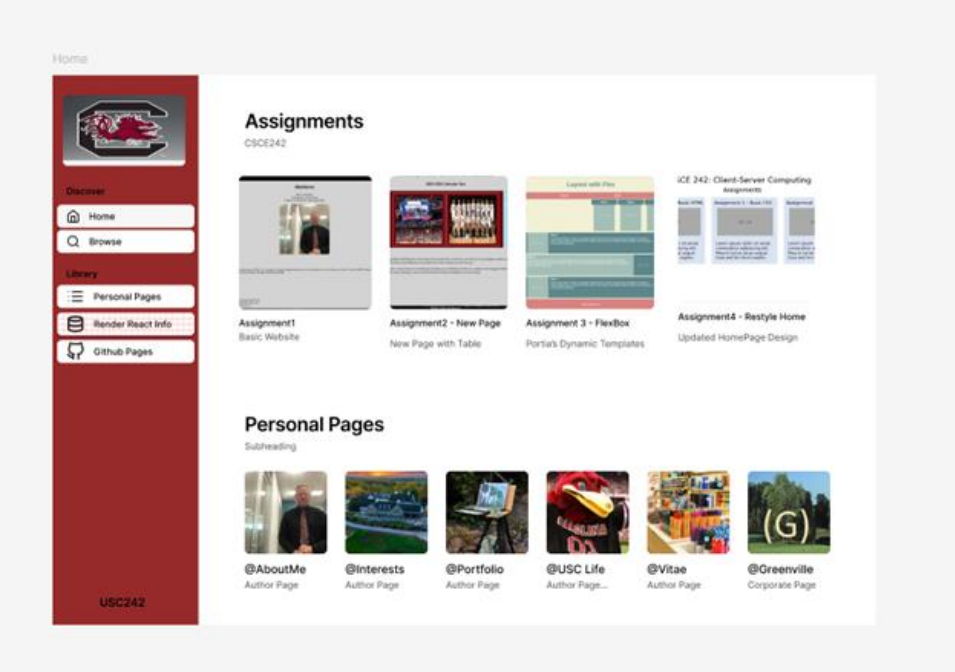
Other than the Home Page:

There are three main Navigation Pages:
Personal Pages
Assignments
Projects

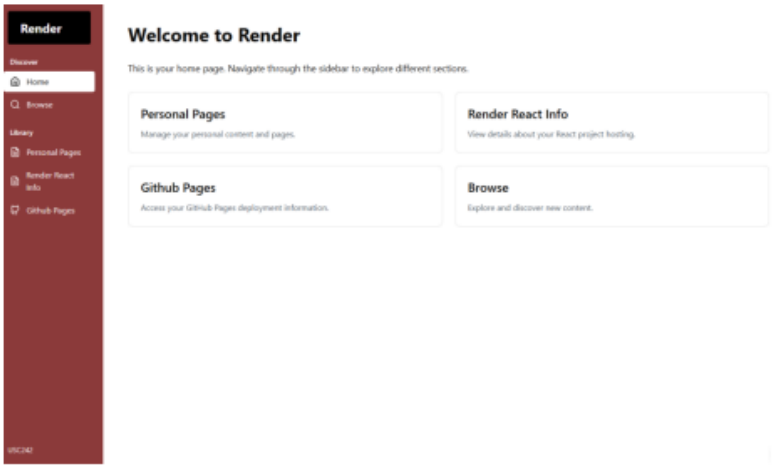
And Two Infrastructure Pages:
Render Hosting
GitHub Devops

2.5 Layout of the Home Main Page

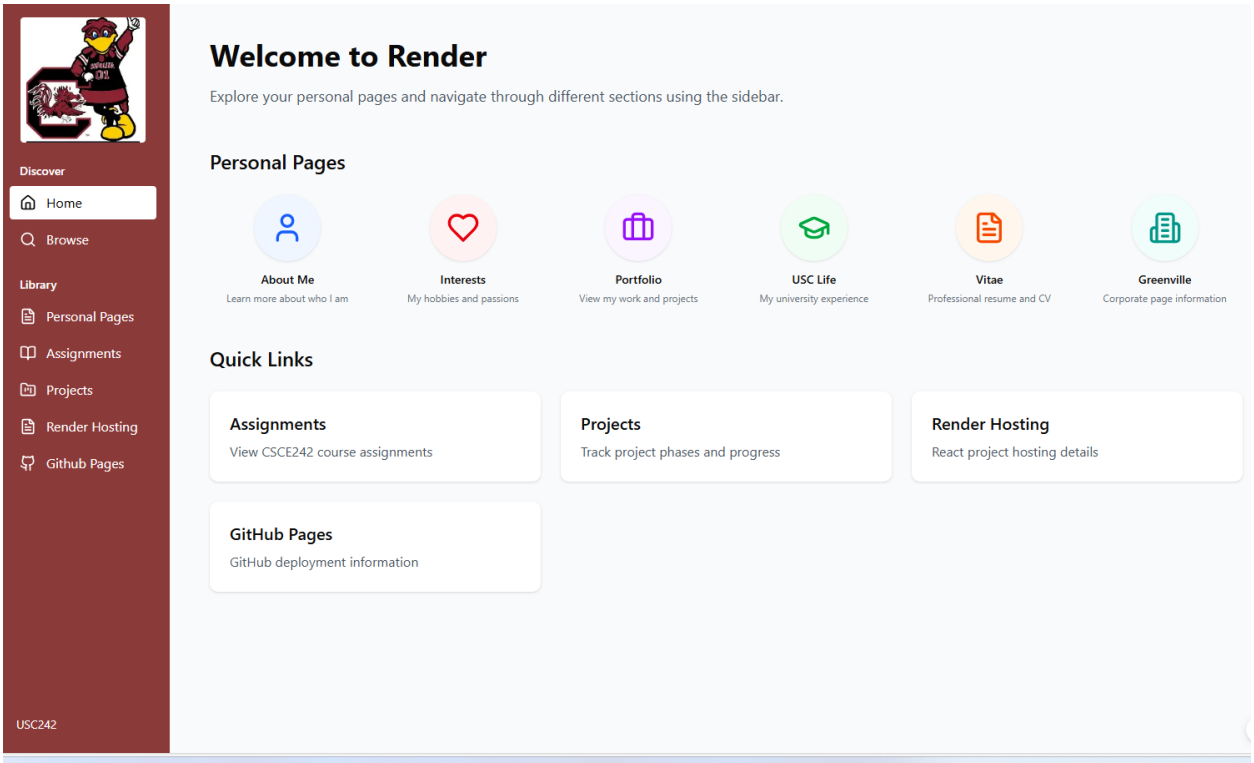
The envisioned home page would only have Active Assignments as seen in a Mongo DB, and all the personal pages.



AND THE AI VERSION OF THE HOME PAGE IN CLAUDE AFTER 1 PASS



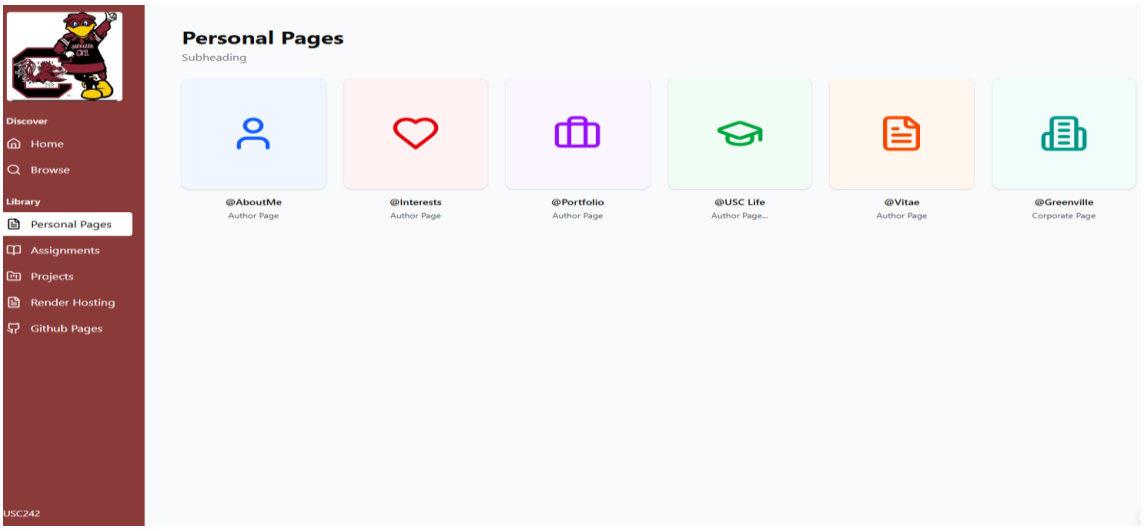
The Home Page Layout in Figma with Claude Enhanced Instructions (Pass 2)



We ended with a Beautiful layout which has Personal Links and Quick Links which aren't quite on target, but with working Navigation and Sub Links on real code.

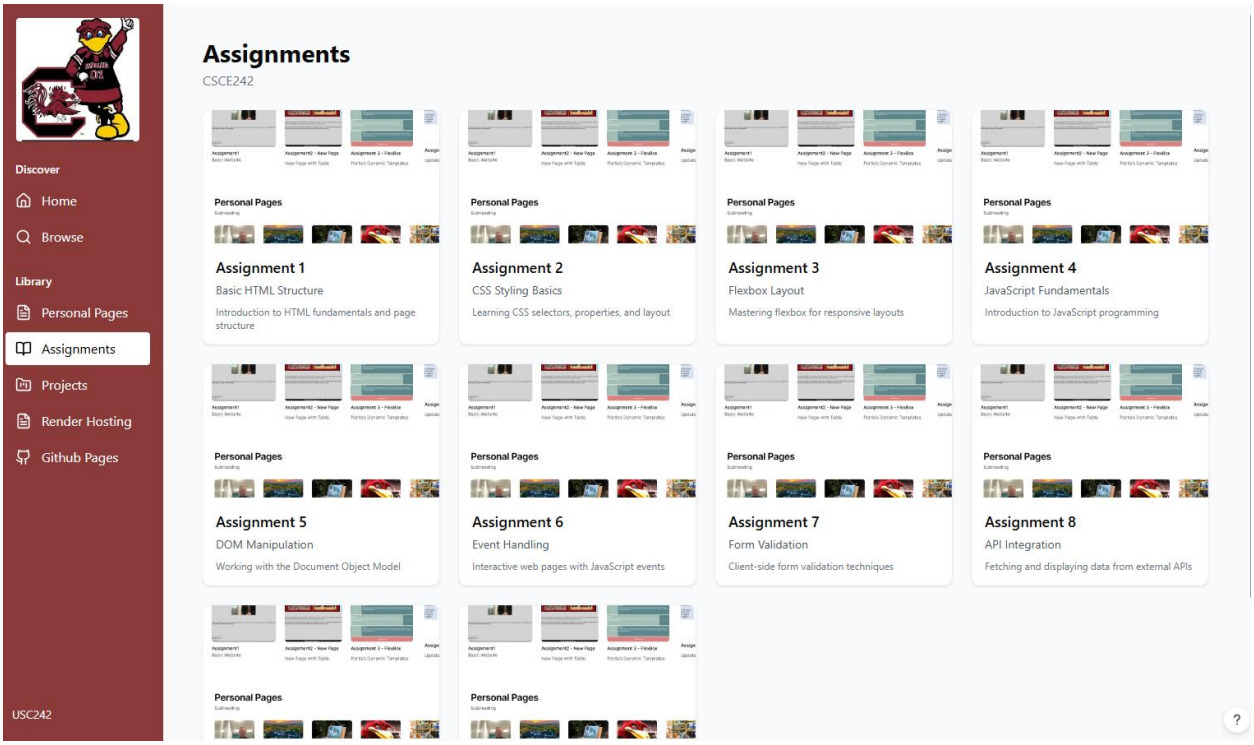
2.6 Layout of the Personal Main Page

We only developed the Personal Page after doing so in Figma Make and didn't go back to update the original board which is redundant. We had to prompt Claude to generate Icons which were attractive. However after we got this result we believe our original design was nicer than what Claude delivered from a design perspective. Claude's pages are however in React form and easy to fix.



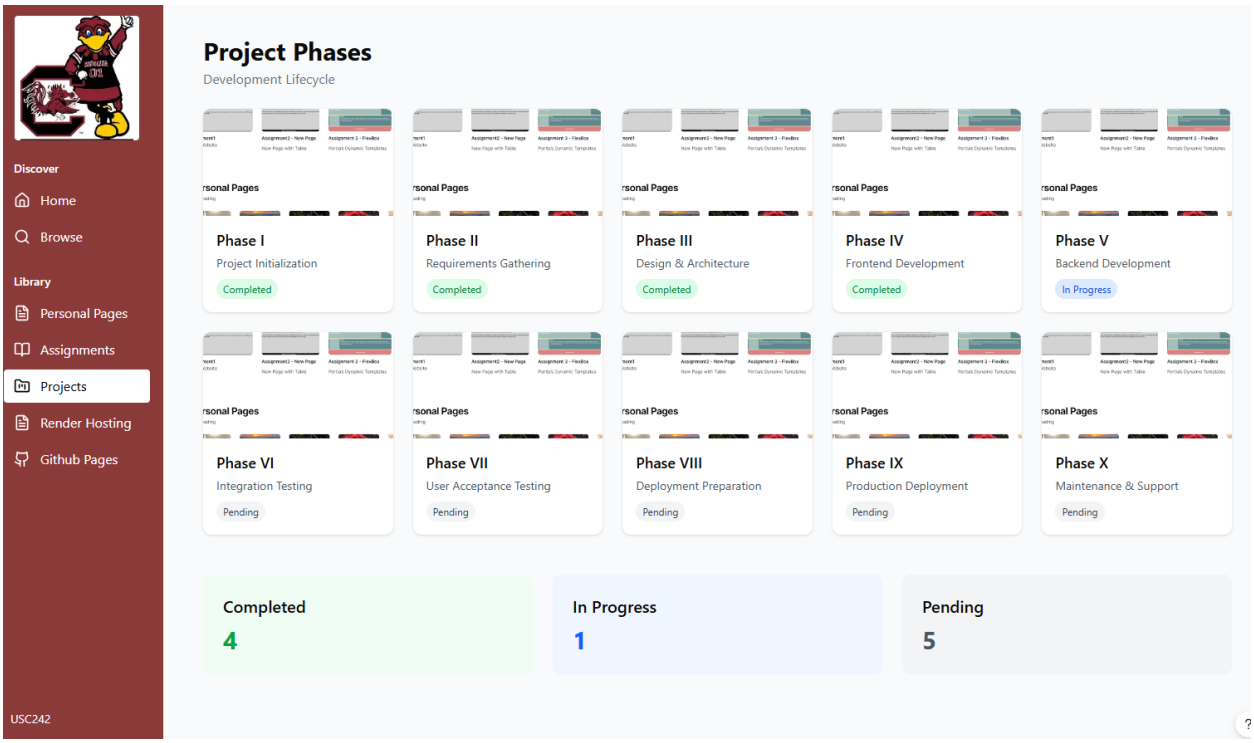
2.7 Layout of the Assignments Main Page

Claude's Layout of the Assignment Page includes 10 Real Sub-Pages which used up our credits, but this assignment is mostly complete in React and we have working code to deploy very soon so it was a helpful corporate exercise. This site can be rebranded by simply changing the Nav Pages Icons, and changing the color of the side bar.

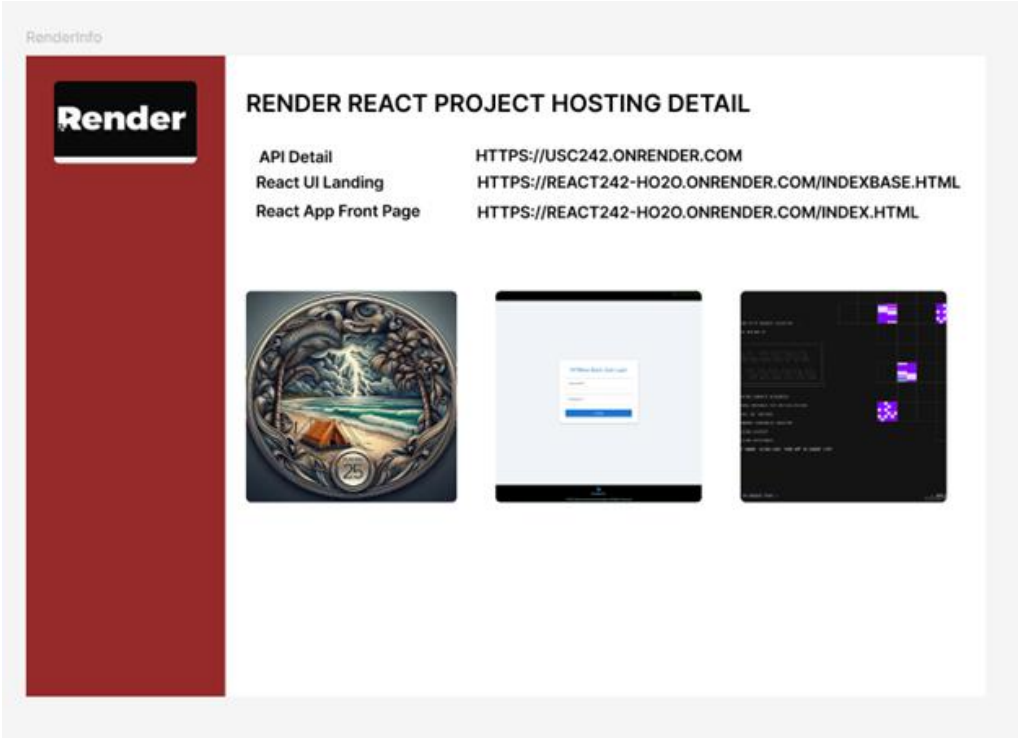


2.8 Layout of the Projects Main Page

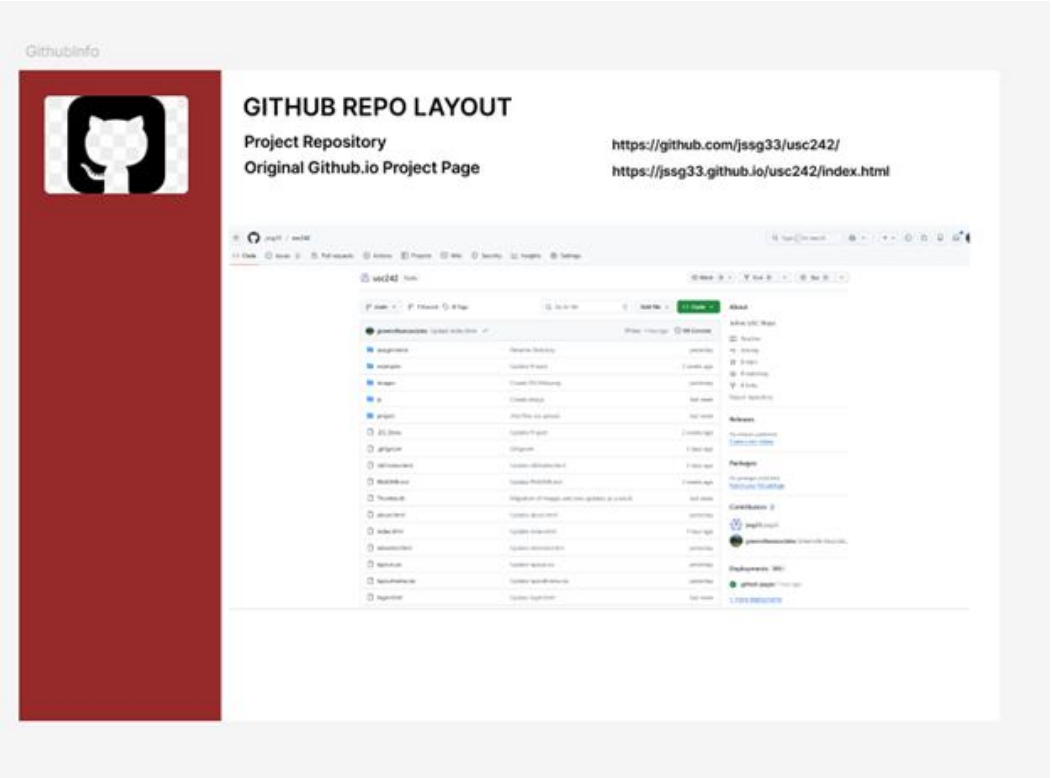
Claude did a much better design that we expected for the Project Phases, setting a completion status, and building a real-time dashboard below.



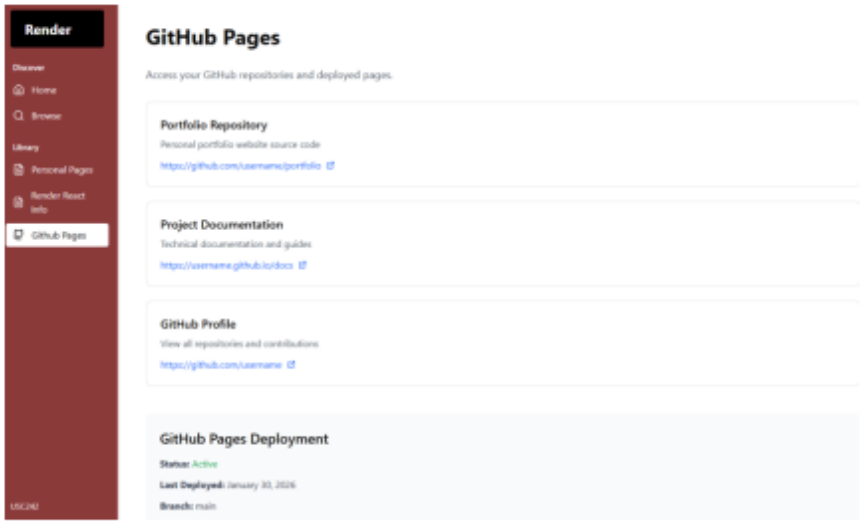
2.9 Layout of the Render Page.



2.10 Layout of the Github Page.



AND THE AI VERSION OF THE GITHUB PAGE BELOW



2.11 FEEDBACK FROM THE INSTRUCTOR

The instructor believed we had misunderstood and needed to submit a wirefame which matched our project description which we submitted in Phase I. This means most of the work we did for this section will be in excess of project requirements but it is useful in practice for Greenville Associates Consulting as a general tool which we need for most projects.

The submission made in Section 3 represents the changed requirements.

2.12 CHANGED SCOPE OF WORK

At the completion of Section2, and After completing the production of most of the Code in react, we have moved to trying to complete the same thing for our original project in building a License Manager.

2.13 SECONDARY REVIEW WITH THE INSTRUCTOR ON 02/06/2026 JUST DAYS BEFORE DUE DATE

The instructor reviewed our plan for the LicenseManager, and she believed it needed to include some sales data to be compliant with the scope of the project submitted. In other words, adding one or more pages of marketing collateral to enhanced the content seen in Section 4.0

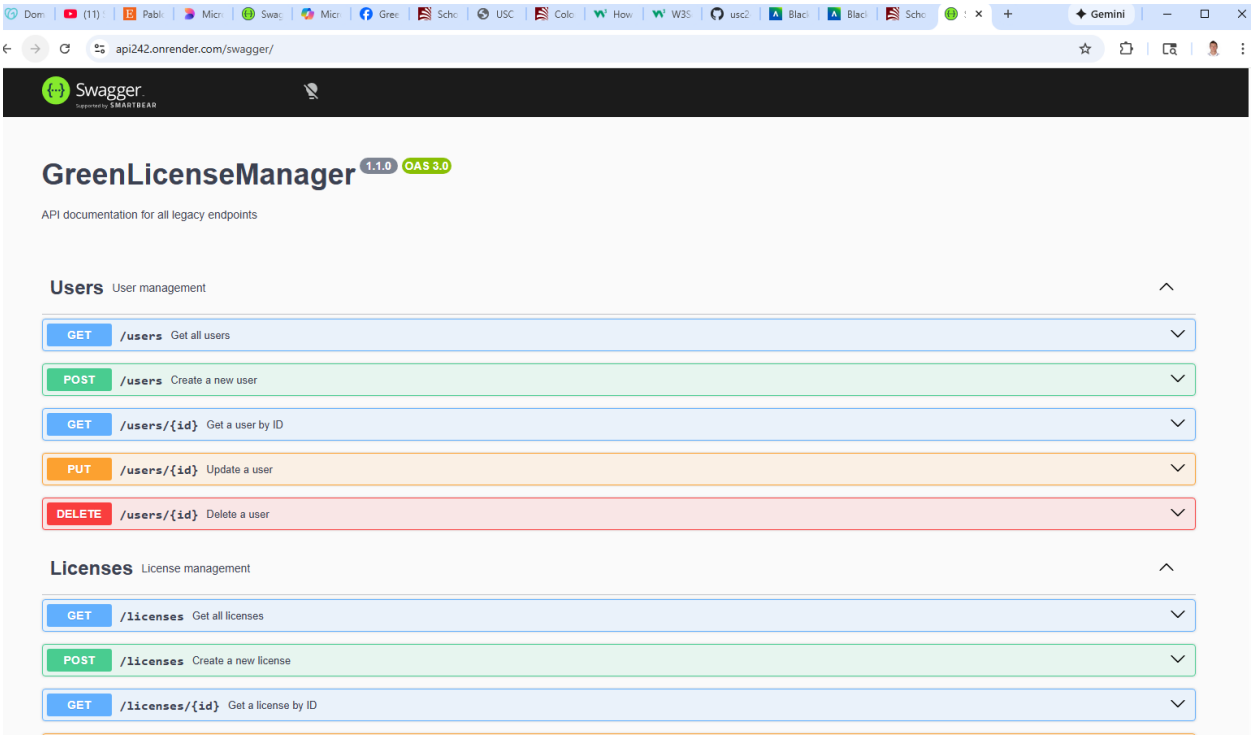
3.0 RENDER APIS IN PRODUCTION

3.1 Executive Summary

In our scope of work for the project we discussed building a 30+ table backoffice as we believed that was the minimal function design after completing 547, 590, 567, 587, and 498(1)/498(2). This includes User Mode Functions, Security, Cart Services, Enterprise Tools, and the specific module under development (Parks, ITIL Service Manager, Solution X, Solution Y). A 30 Table Backoffice without any prior work in this area is a significant design and development burden in itself. It is another thing to put in production in C# APIs, Host the API in a responsible manner, and or build Express NodeJS equivalents.

3.2 Porting And New Code from Azure and Google Cloud to Render

The instructor determined her target for our environment was student accounts on Render, and Mongo DB databases connected to Express. This environment is now complete and in production. Initially we built and checked our work without swagger/OpenAPI, then asked various AI tools if it is supported on Express, and since it is we felt great to get to more or less the same code design for our backoffice supportable on NodeJS as it is on Microsoft Hosting platforms as DotNet Executables. Unlike Microsoft Tools, The Render Environment can actually run Yarn, and Vite/NPM development scripts to build the source code and is superior from a production environment as it doesn't require 16GB of RAM to Run Vscode/Visual Studio. A Beautiful Solution.



3.3 SCHEMA DESIGN FOR THE PROJECT

The current schema which we believe encompasses all the requirements of this project is approximately 31-40 tables on completion which is large project for a single developer by itself. We believe this is only possible as we have been working on a target operating schema for more than 18 months in 6 other classes. We advised the instructor that a 5-6 page project is doable for a single developer in a 2XX class but is not anywhere near production quality deliverables of 547/590 built by teams.

Nevertheless here is our proposed solution.

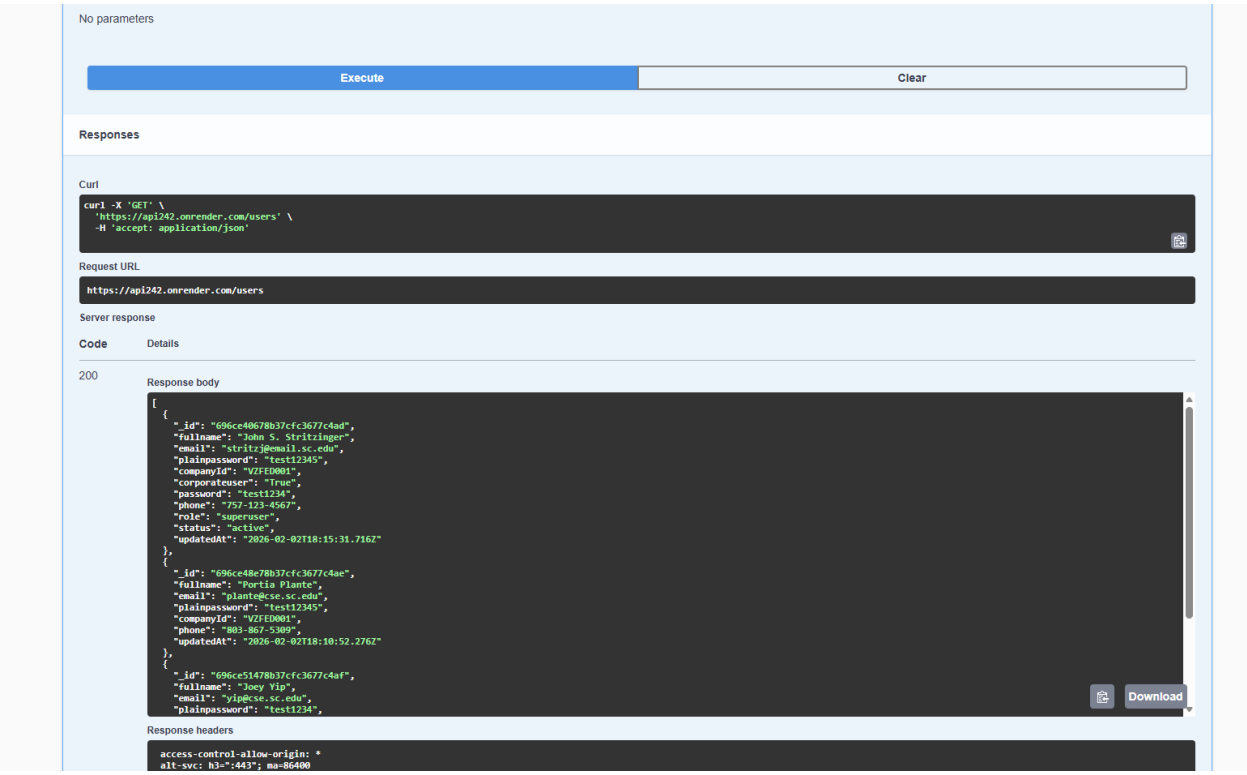
```
// -------------------------
// ROUTES
// -------------------------

// Users & Accounts (4)
app.use("/users", require("./routes/userRoutes"));
app.use("/api/notices", require("./routes/userNoticeRoutes"));
app.use("/usercontacts", require("./routes/userContactRoutes"));
app.use("/userhelp", require("./routes/userHelpRoutes"));

// Companies & Branches (3)
app.use("/companies", require("./routes/companyRoutes"));
app.use("/branches", require("./routes/branchRoutes"));
app.use("/instances", require("./routes/instanceRoutes"));

// Logging & Security (5)
app.use("/api/apilogs", require("./routes/apiLogRoutes"));
app.use("/api/adminlogs", require("./routes/adminLogRoutes"));
app.use("/downloadlogs", require("./routes/downloadLogRoutes"));
app.use("/licenselogs", require("./routes/licenseLogRoutes"));
app.use("/userlogs", require("./routes/userLogRoutes"));

// Licensing (1)
app.use("/licenses", require("./routes/licenseRoutes"));

// Parks (1)
app.use("/parks", require("./routes/parkRoutes"));

// DTOs (2)
app.use("/api/GCPARKS", require("./routes/gcParksRoutes"));
app.use("/api/gcparks", require("./routes/gcParksRoutes")); // lowercase alias
app.use("/api/CGCART", require("./routes/cgCartRoutes"));

// Batch Processing (1)
app.use("/api/batches", require("./routes/batchRoutes"));

// Commerce / Sales (7)
app.use("/products", require("./routes/productRoutes"));
app.use("/reviews", require("./routes/reviewRoutes"));
app.use("/salescatalogue", require("./routes/salesCatalogueRoutes"));
app.use("/invoices", require("./routes/invoiceRoutes"));
app.use("/invoicelineitems", require("./routes/invoiceLineItemRoutes"));
app.use("/payments", require("./routes/paymentRoutes"));
app.use("/refunds", require("./routes/refundRoutes"));

// Cart System (3)
app.use("/cart", require("./routes/cartRoutes"));
app.use("/cartmaster", require("./routes/cartMasterRoutes"));
app.use("/cartitems", require("./routes/cartItemRoutes"));

// Reservations (1)
app.use("/reservations", require("./routes/reservationRoutes"));
```

```
// Cards (1)
app.use("/cards", require("./routes/cardsRoutes"));

// Scopes & Project Tasks (2)
app.use("/scopes", require("./routes/scopeRoutes"));
app.use("/projecttasks", require("./routes/projectTaskRoutes"));

// -------------------------
// TOTAL ROUTES: 31
// -----
```

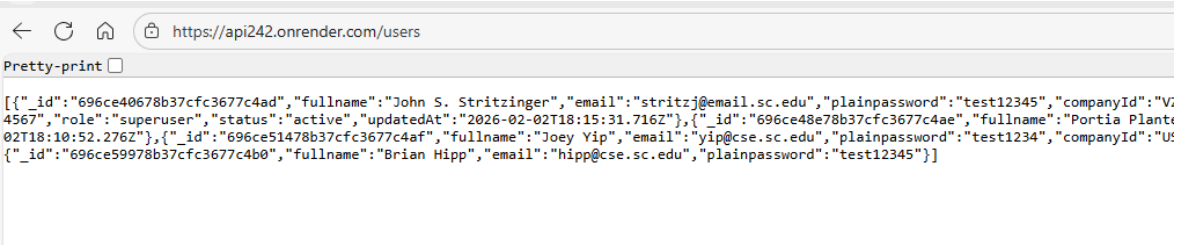
THESE ALL SUPPPORT GET, AND PUT, AND GET, PUT AND DELETE BY ID OUT OF THE BOX (5 SETS OF ENDPOINTS FOR EACH TABLE). This is basic SQL support, and does not support additional items like capacity checks, user notification, and backoffice services which may need additional tables, and lots more Endpoint logic none of which are in scope for this class.

3.4 SWAGGER AND EXAMPLE OUTPUT FOR THE PROJECT – USERS

Swagger is returning the User Table Below



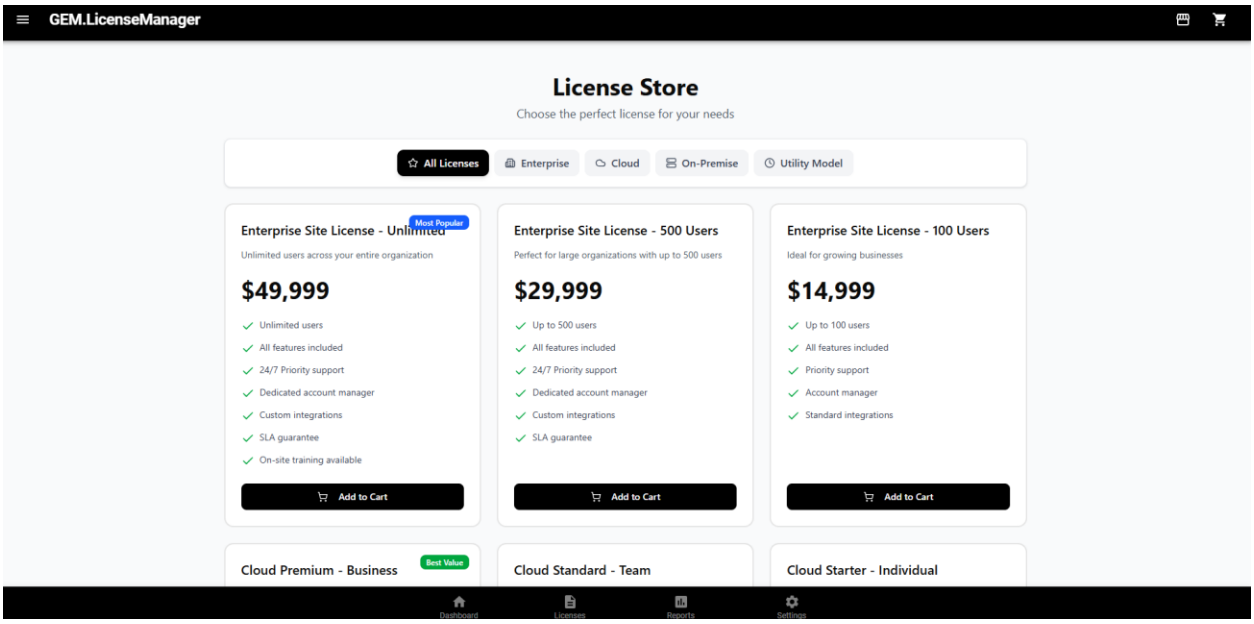
3.5 DIRECT ACCESS TO THE API FROM AN APPLICATION RETURNS A JSON



SECTION 4.0 HOSTING

We are hosting live applications on Render, and on Figma Make for the project. Links are on the Github.io page for the class.

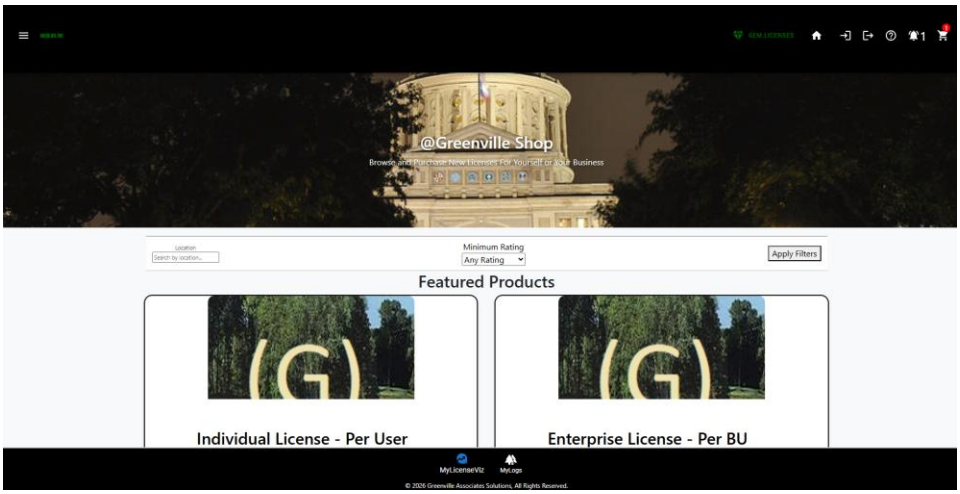
4.1 Render Hosting of Figma Make Output



4.2 Render Hosting of John Stritzinger's Enhanced Project Code – CG UI Design

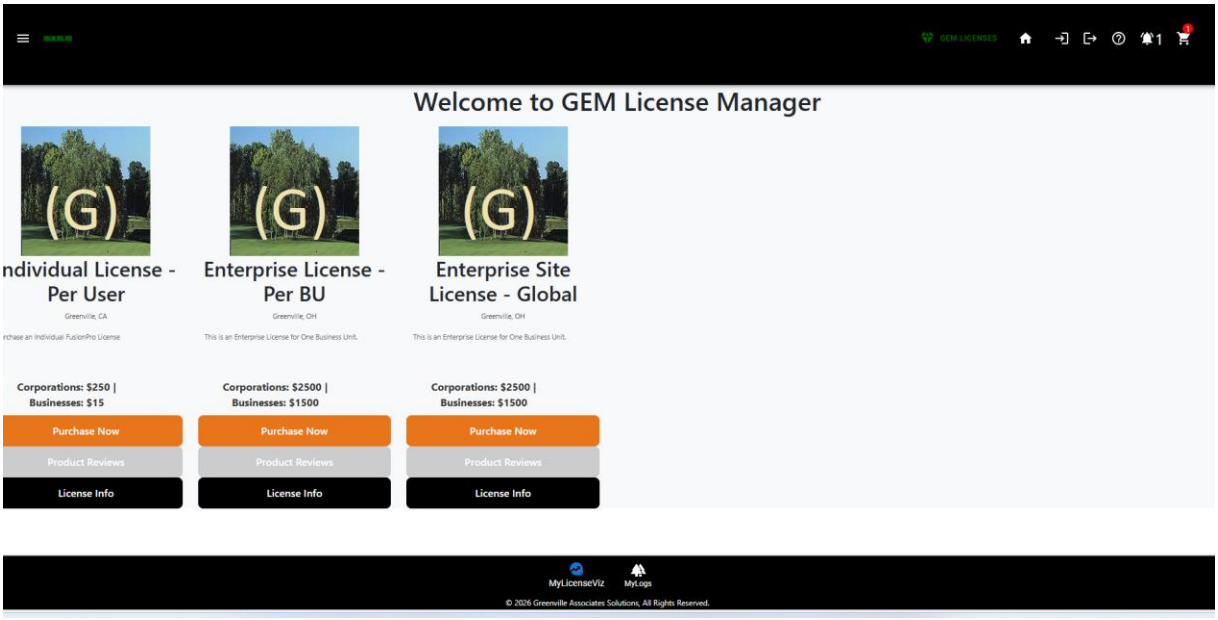
We have built a Target Environment on React Material UI pages which is also hosted and nearing completion within a few weeks more than two months in advance. The layout of this page was designed by Capgemeni Consulting

Enclosed is view of the home page.

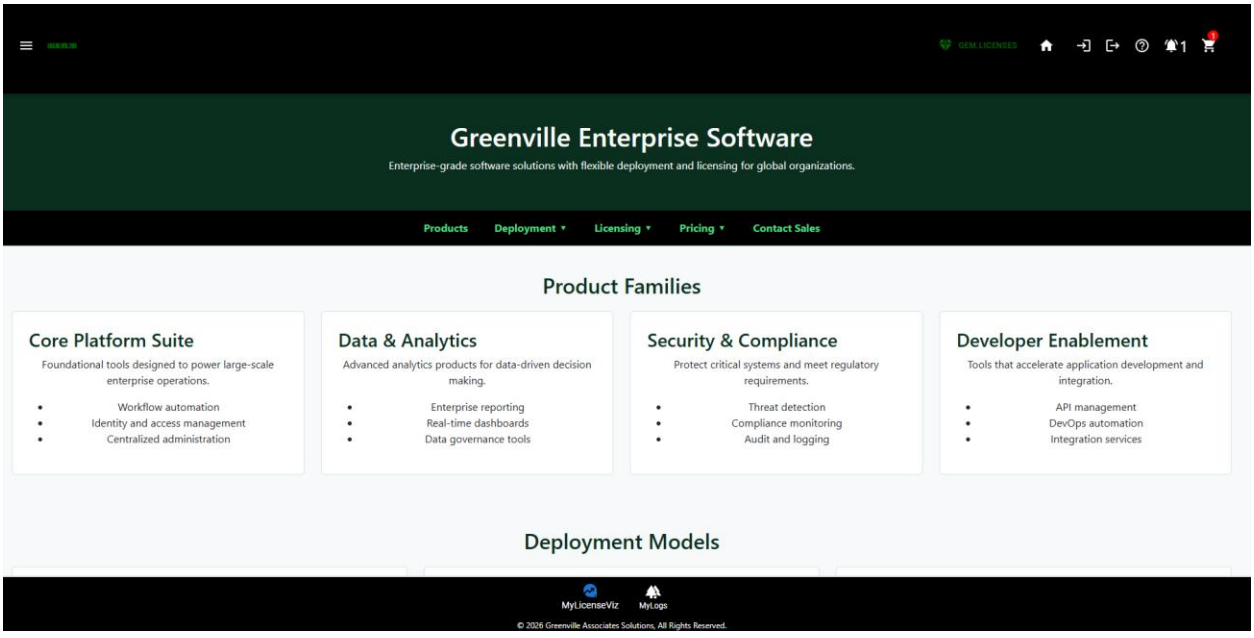


4.3 Render Hosting of John Stritzinger's Enhanced Project Code – JSS Redesign

At the present time, we believe the CapGemeni Site Layout is more attractive than ours in either potential variant sent below. However we are moving to Normalizing the design between the sites as both are deficient now, as the Search Bar, and Featured Products pages are breaking the mobile environment on smaller screens.



Enclosed is the Enhanced Layout after Instructor Feedback…but it is not quite right either… completing the design to collective happiness will take a bit longer.



4.4 Live Database Queries Working:

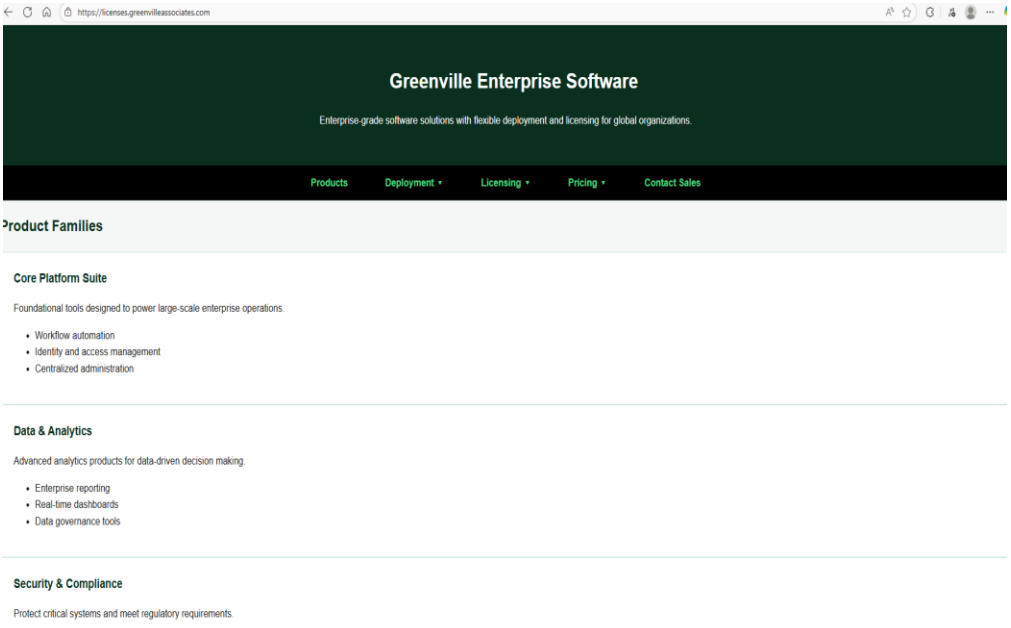
a) Product Pricing
b) License Types
c) Sales Information
d) User SSO
e) User Notices
f) User Logs
g) User Notices
h) User Help Tickets.

We have working Queries for most of the logical requirements for the program. All of this code is new code.

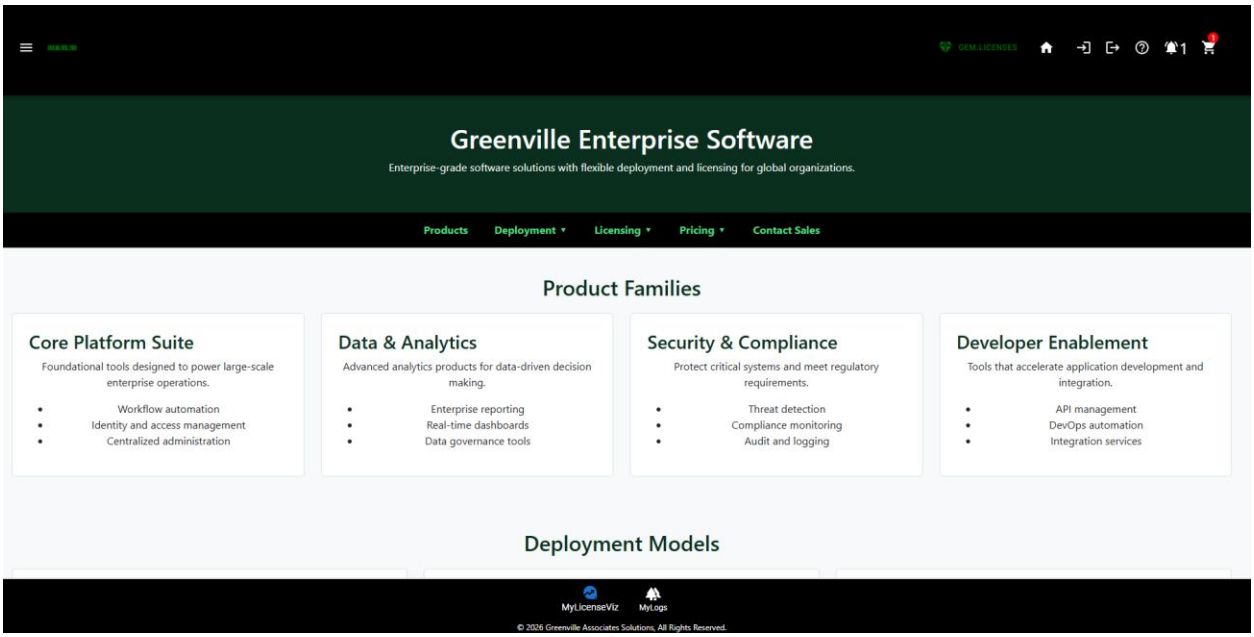
4.5 Advantages

The previous project with CapGemeni Delivered a New Cart, But with AI assistance we were able to build a new model which works up to the post. We think with some new guidance we can get the Post to work with AI in a few steps. But it requires more than 10 parameters in an AI session including working target JSONs. It's a non-trivial exercise with AI, and without AI we have two other working cart architectures in production we can port to this assignment. This includes the three tiered cart model we already presented.

4.6 Target HTML/JS Site Up on licenses.greenvilleassociates.com



5.0 TARGET GREENVILLE LICENSE MANAGER WIREFRAMES – COMPLETE



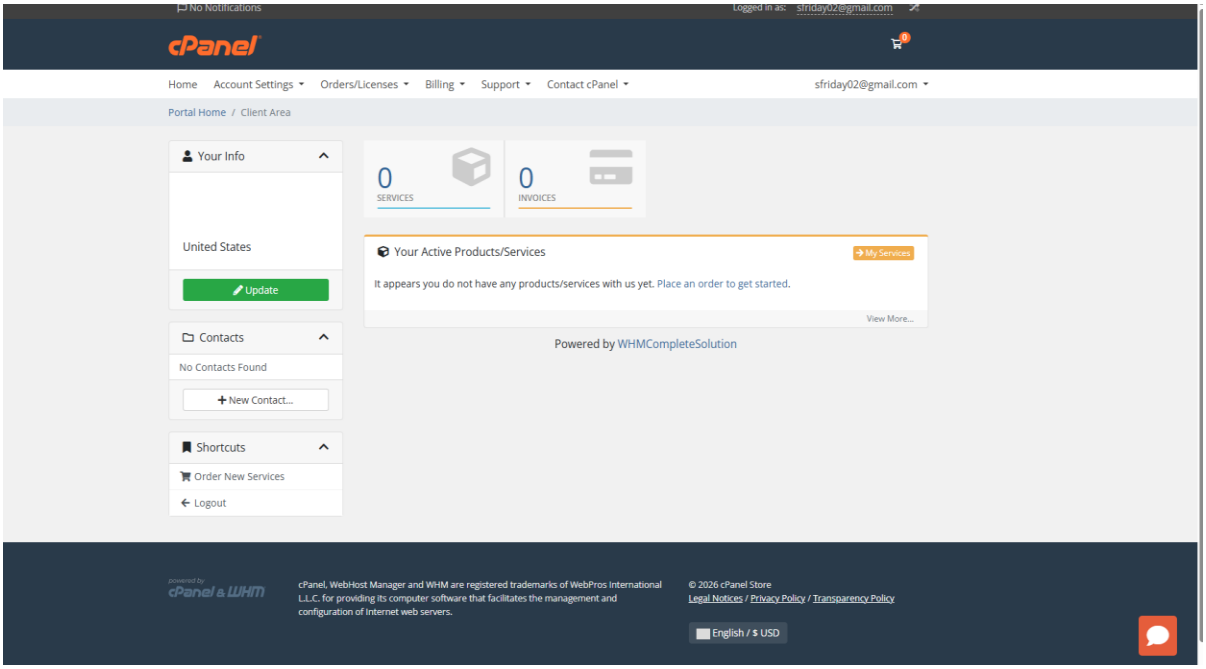
5.1 Executive Overview – Translating Scope of Work to Target Designs.

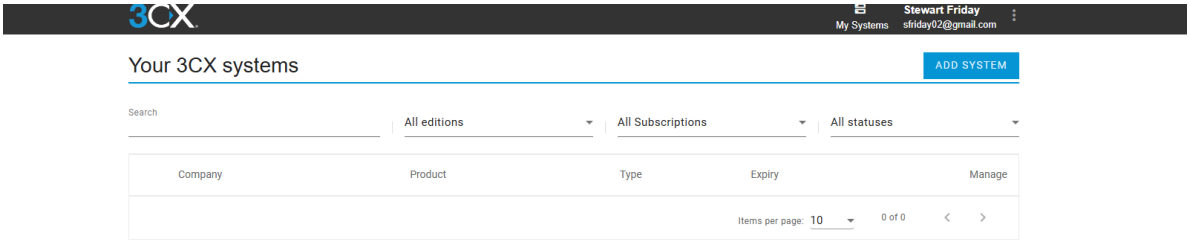
We have sought to build a Landing page for our LicenseManager which discusses what types of licenses we sell in the Greenville Enterprise Software(GES) Division of our company. The idea is after we logged in we would only see domain specific information

5.2 Competitive Analysis

Other competing products from 3CX VOIP tools, and cPanel which we think both do an excellent job handling Cloud Licenses and we look at their license managers as the gold standard which we are not trying to copy, but we think we can improve it substantially in the following ways:

a) Our competitors tool only allows us to view our current licenses, add more and upgrade existing ones. This is significant code.
b) Our competitors tools allows us to manage our business accounts, and purchase history. This is previous sales history. This is similar to restaurant companies like Sonic and Ihop restaurants which show your previous transactions online.





5.3 Significant Improvements over Competing Designs

The four competitors software tools we mentioned only show a single user or companies licenses. They have no view of enterprise licenses, or in a wholesale model the usage of agents. Furthermore these tools have no Administrative equivalents which show dashboard, and usage of platforms, although we think that our competitors have this information externally for investor purposes and for understanding the usage of their products in more detail.

We intend to provide aggregated information to Hosting Providers who have 5, 10, 100, or hundreds of containers of our software across multiple hosting regions, grids, and hybrid clouds.

We intend to use advanced logging and security tools to extend our SSO framework to a global platform and capability across our Global Managed Grid Infrastructure.

For the same reason they have no ability to provide any enterprise usage reports for their customers without professional services rollouts. We had intended to mirror these very successful companies deployments of license management which is superior even to Cisco or Microsoft in our opinion but she told us she thought it wasn't very good.

5.4 Structure of Final Workflow includes Up to 18 Pages All of Which are Designed and in near working form.

Major structures of our document include a TopBar, a BottomBar, and interior Mid-Page Navigation as seen in the diagram in 5.0 using Material-UI Tool bars. These engage the React-Router.

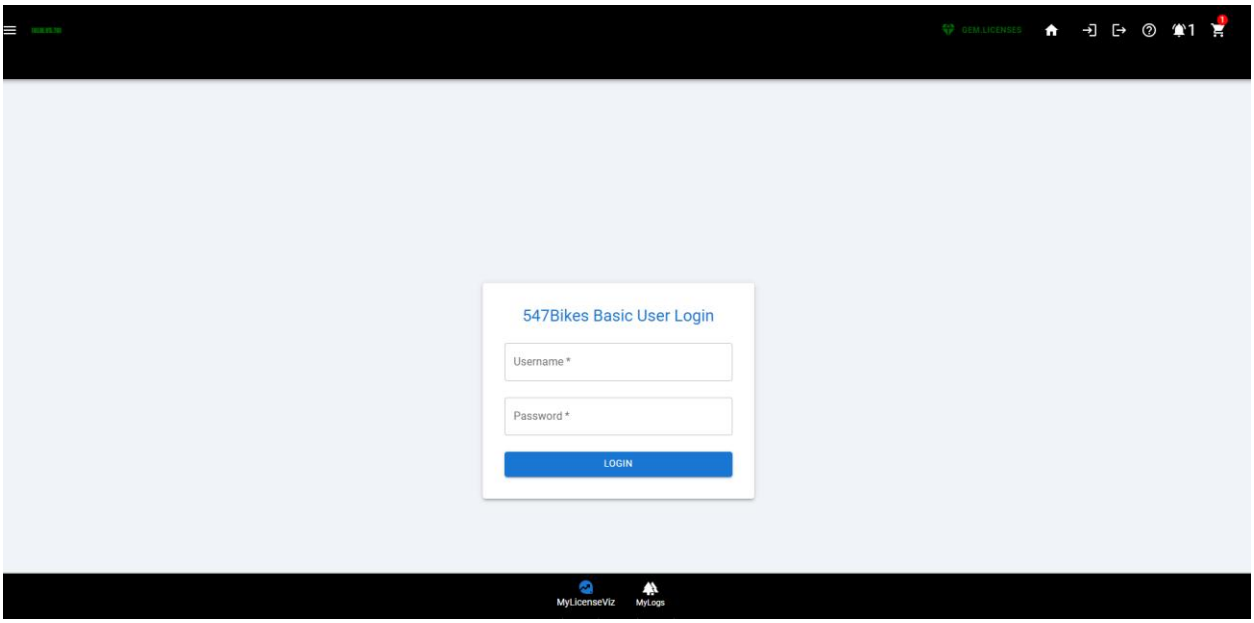
5.5 Menu Options DropDown & Pages Accessible via Top Nav Bar

We intend to load the Sales Site as our Landing Page for the License Page with a General overview of the License Types available and where they can be used. Furthermore, the Login(1), Logout(2), UserNotices(3), UserHelp(4), And Cart(5) can be accessible from the Main Nav Bar and sized to work on all devices. The Menu Bar Supports additional information Welcome(SystemInfo)(6), Signup/Registration(7), All Licenses(8), Corporate Profile(9), ProductReviews(10), LicenseInformation(11), Home2(12), LicenseLogs(13).

The Bottom Nav Bar for End-Users is Operational. It shows Logs, and Current Licenses. On the Current Licenses Tool You can Buy More(14), or Upgrade Licenses(15) with default pricing level loaded. Then there is a cartreview(16), paymentreview(17), and cardmaintenance(18) page environment which is more than 3X the project requirements for the class, and with a much larger backoffice on Mongo DB our project is significantly more sophisticated.

These Screen #'s are included below for more information.

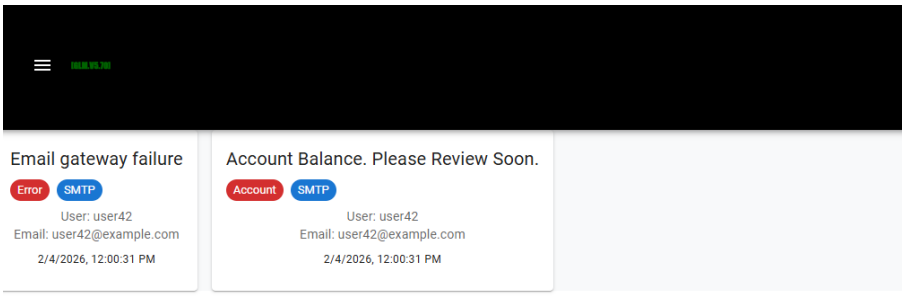
5.16 LOGIN(1)



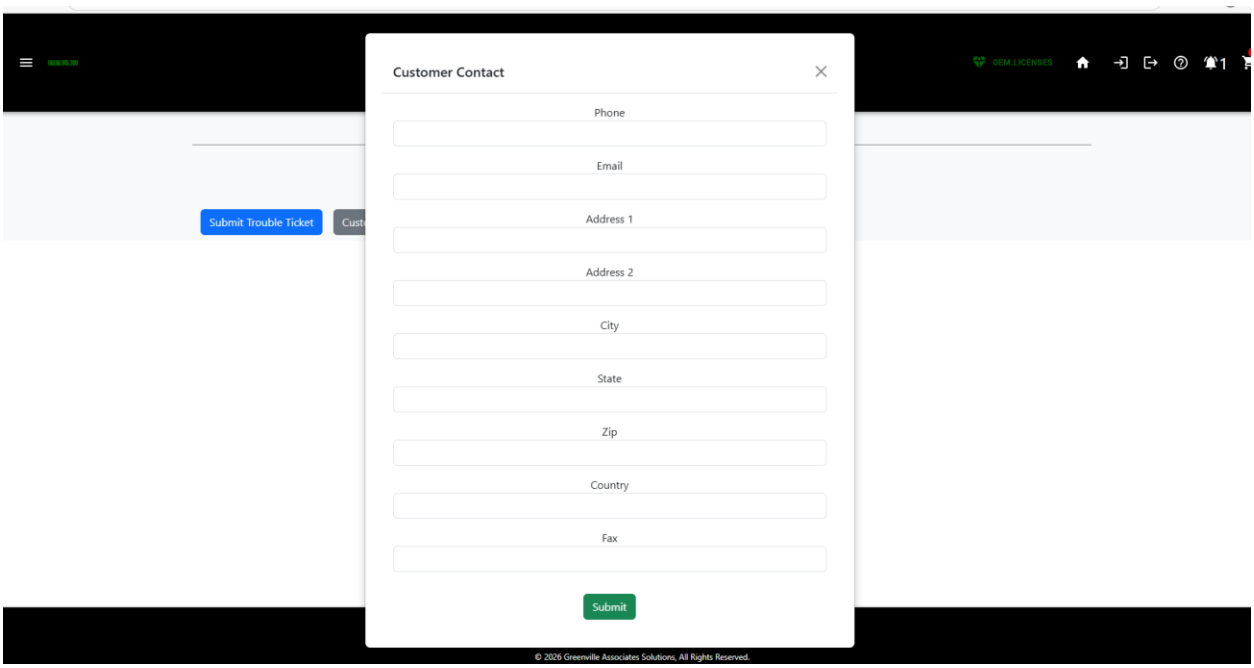
5.16 LOGOUT(2)

Processed in the background with a spinner.

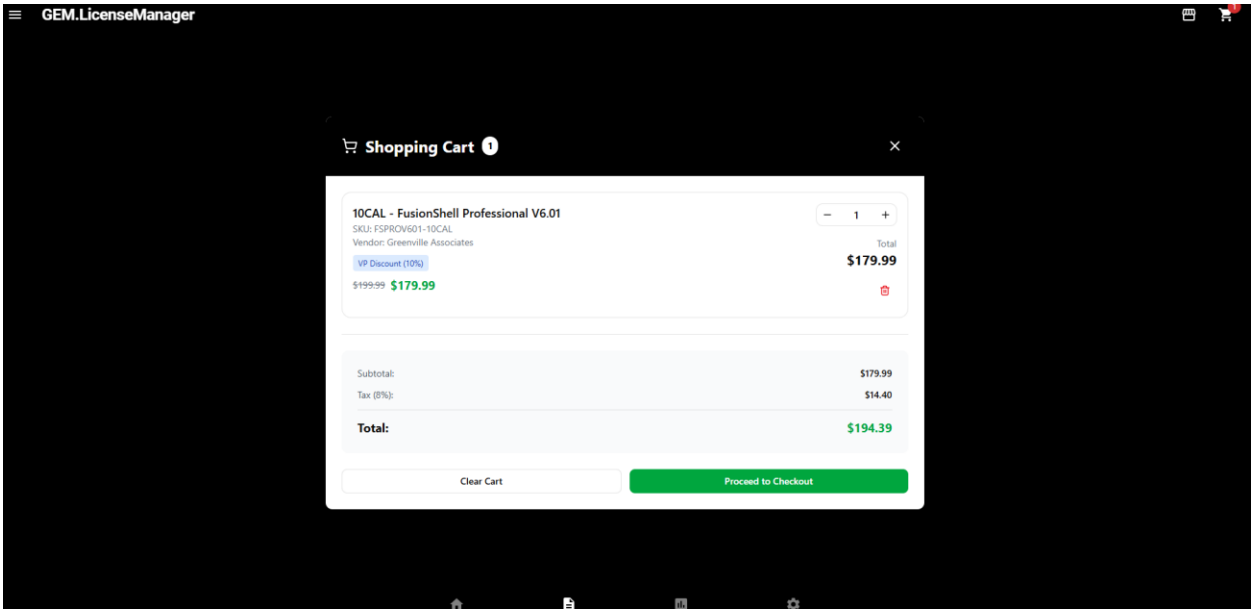
5.17 User Notices(3)



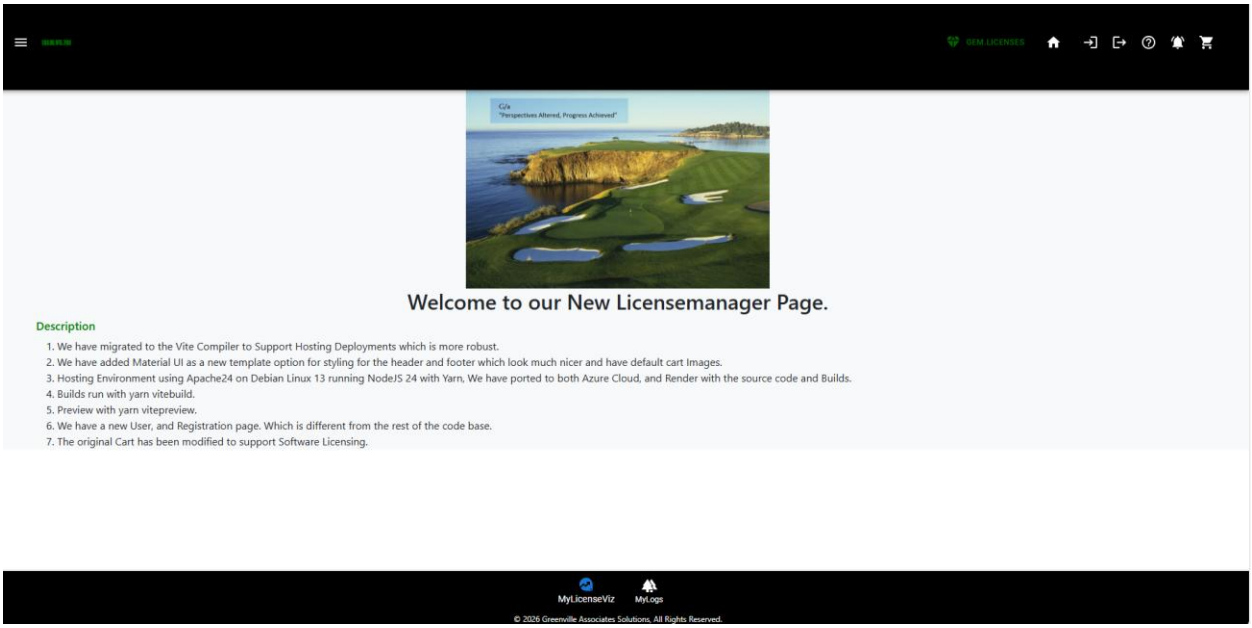
5.17 User Help(4)



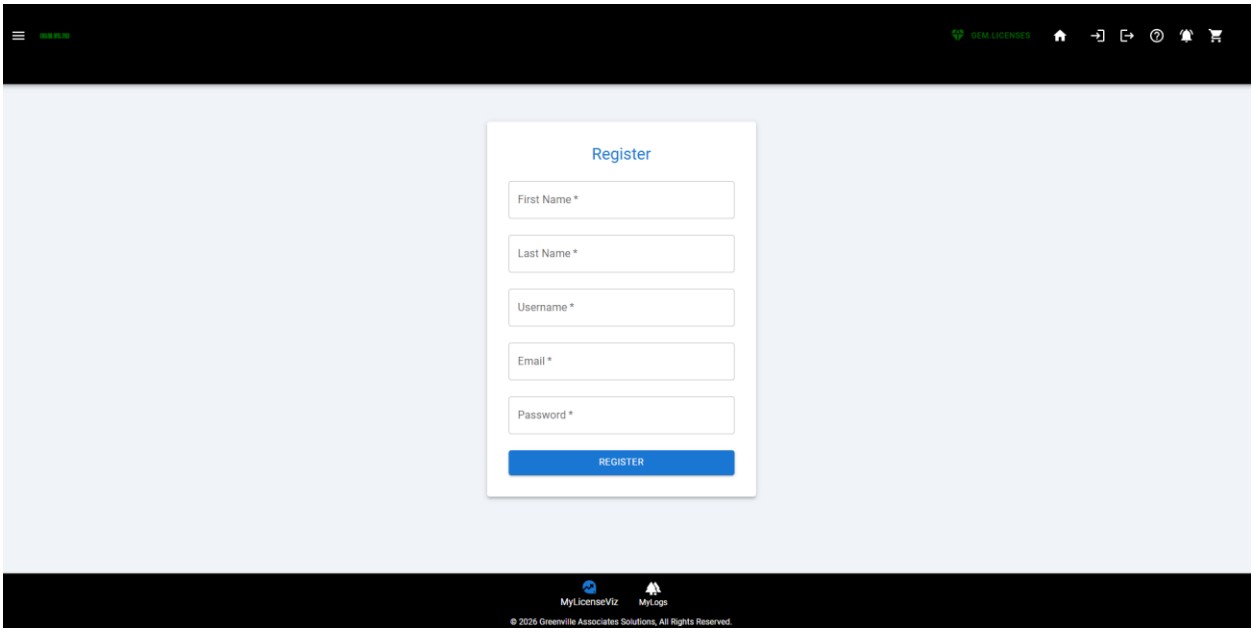
5.18(5)



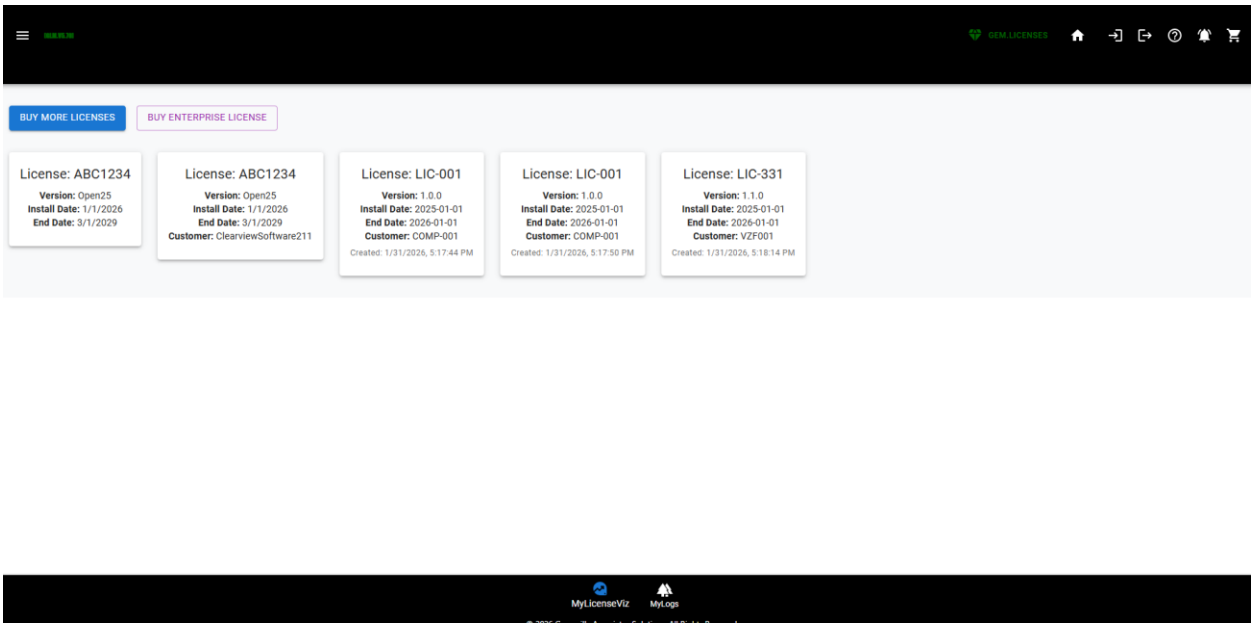
5.19(6) Welcome(SystemInfo)(6)



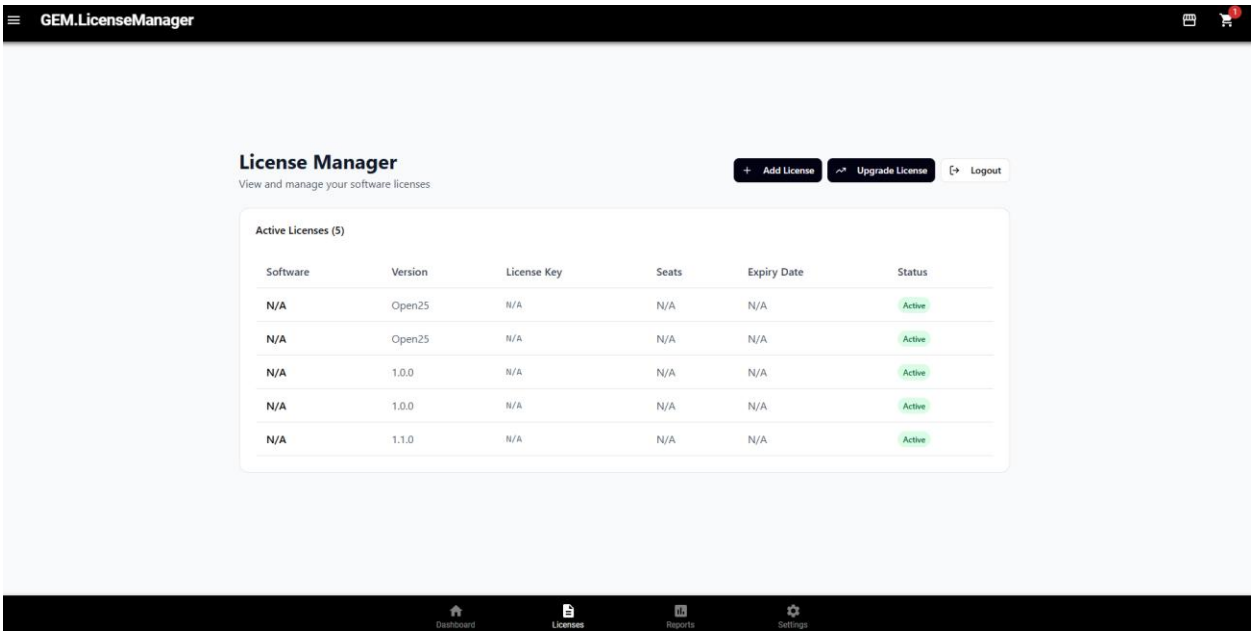
5.20 Signup/Registration(7)



5.21 All Licenses(8) Legacy

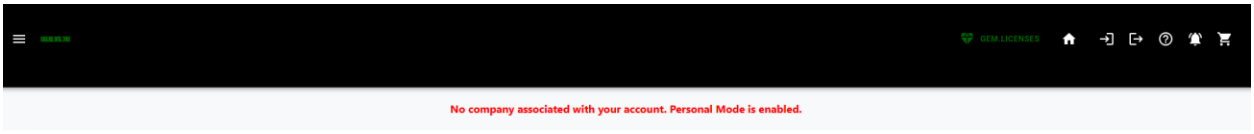


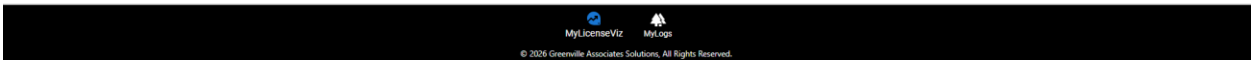
Legacy Licenses – Figma Design(Improvement)



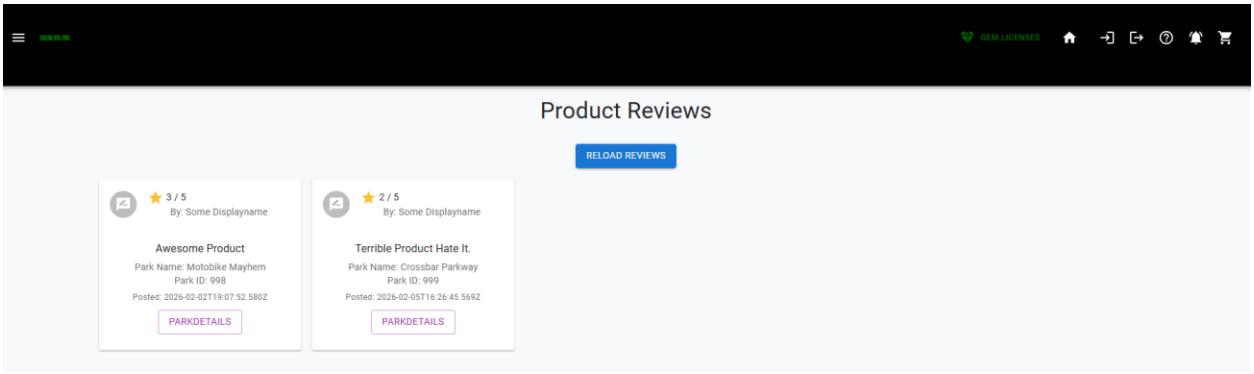
5.22 Corporate Profile(9)

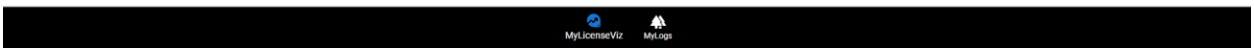
This page is not in a good tangible form….Because I don’t have a JOIN which works in the database so I can debug the result yet…. IE UserProfile to Corporate Account has to work.



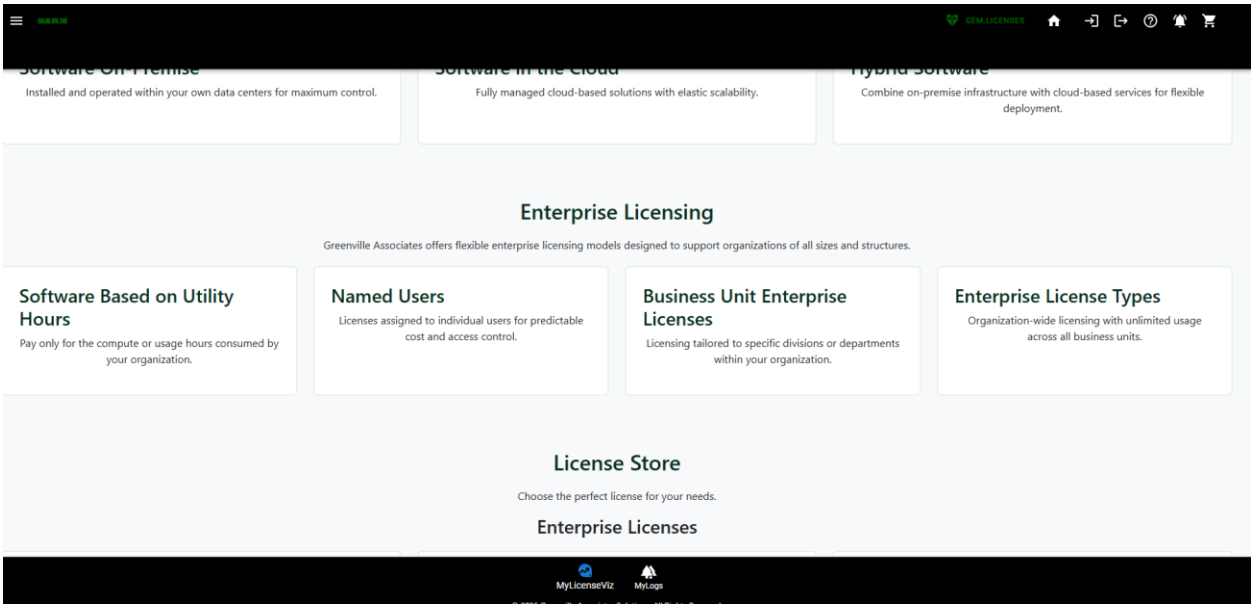


5.23 ProductReviews(10)



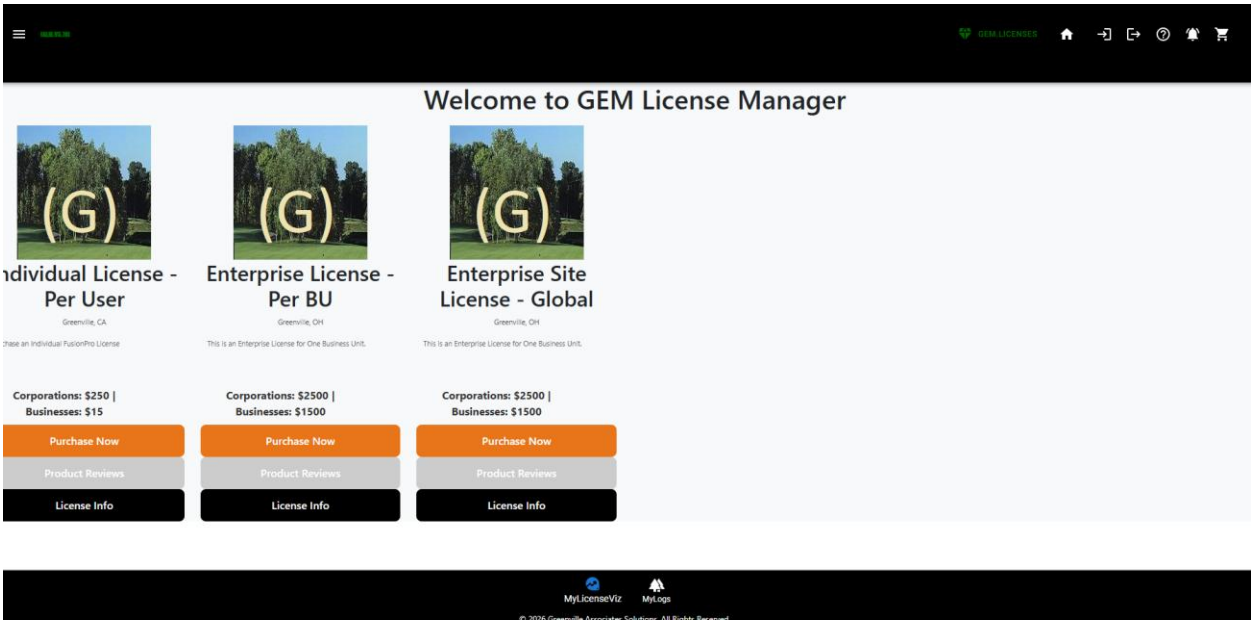


5.24 LicenseInformation(11)

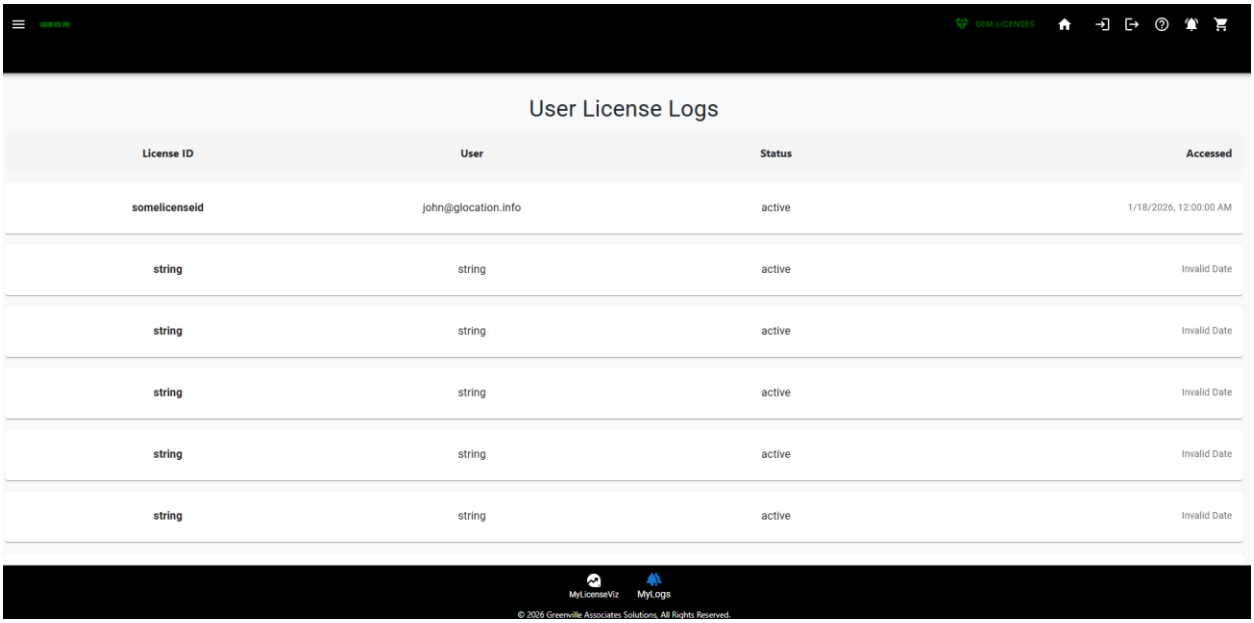


5.25 Home2(12) – Different View of Products – All Products in All conditions.

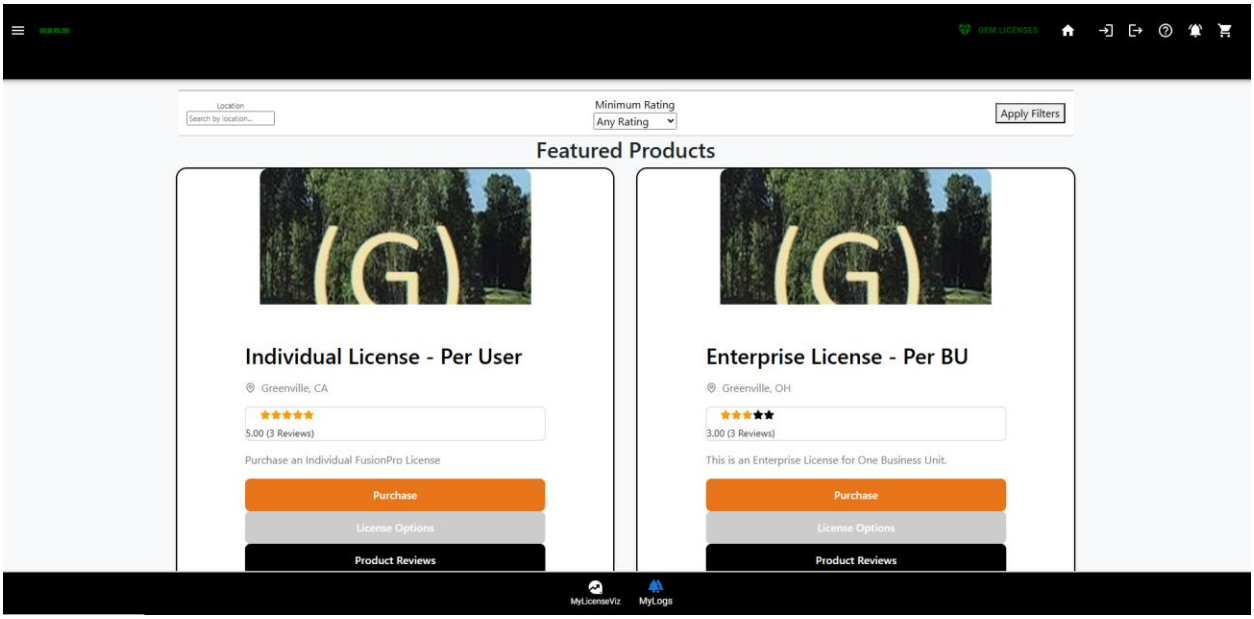
This is a Full List of Products while the home page is supposed to be just a subset of key products.



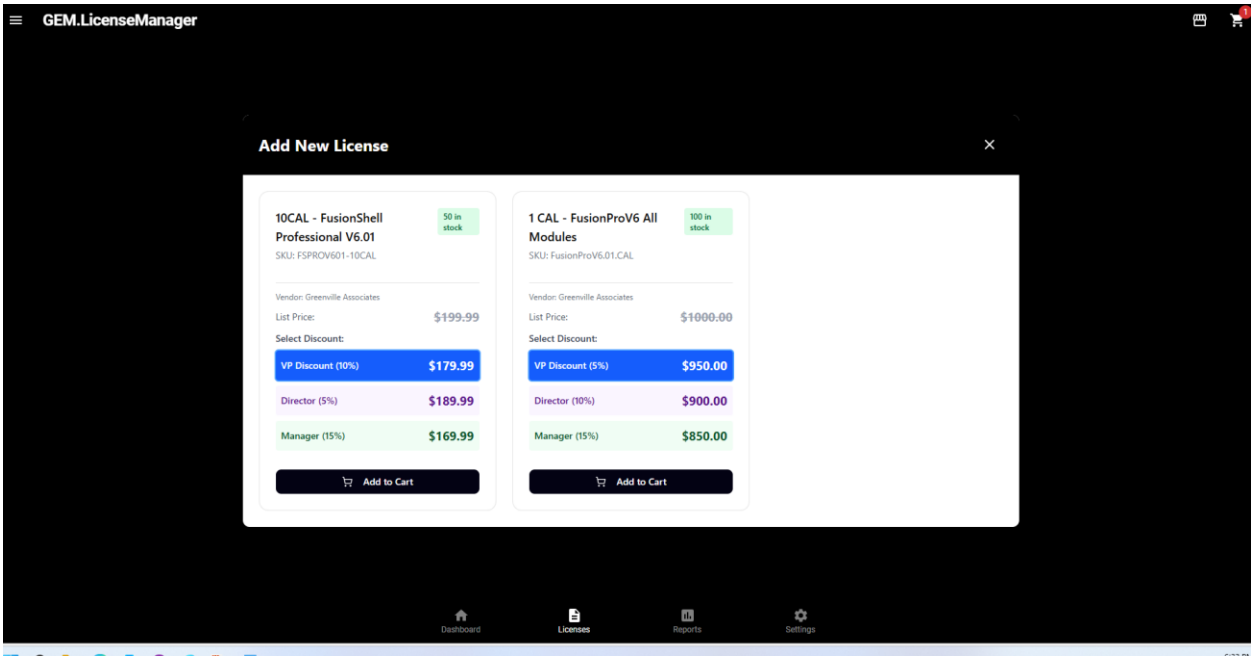
5.26 LicenseLogs(13).



5.27 Buy More(14)



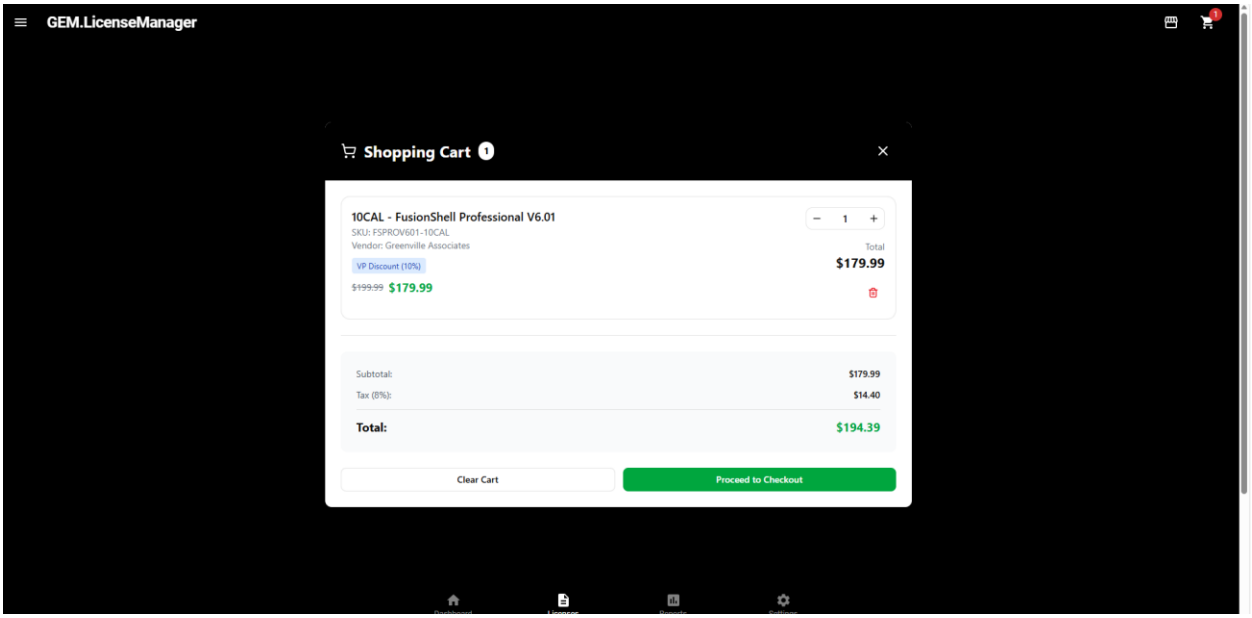
Inside the License Manager



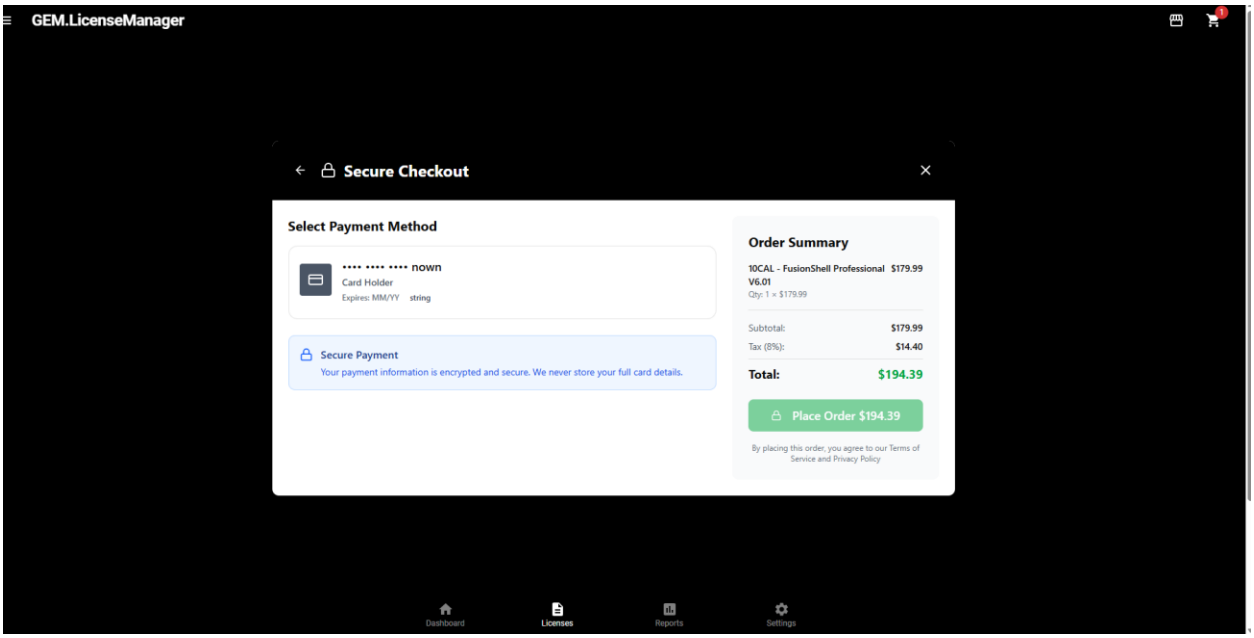
5.28 Upgrade Licenses(15)

Same as Above right now… will eventually filter for only products which can be upgraded.

5.29 Cartreview(16)



5.30 Payment Review(17)



5.31 Card Maintenance(18)

This is implemented in the CG Interface using custom Greenville Components. This button is not implemented in the Figma AI model as we ran out of credits.

6.0 RAPID PROTOTYPING

In this class we have been talking about using Figma to take a Wireframe, and build out code via a Publish Method. We agree that this works, however we can do the same thing using our own code which we pull from a combination of sources which isn't automated by Figma but is automated via production Rapid Prototyping which usually identifies most of the challenges needed in the backoffice.

6.1 Building the API first

When you build the API first in rapid protoyping a JSON is produced from your handiwork as previously demonstrated. Most AI tools can build entire screens in .ASPX, DotNet, C++, HTML/JS, or JAVA from this json alone. Building Wireframes before databases therefore is of questionable value except for learning programming languages.

6.2 Rendering From JSON Before Wireframing using Rapid Development is Advised

When you render pages from rapid development you take the time to look at flows of pages and how they are accessed. You identify tables and data you need to complete your screen which then usually adds layers to setup fields, and data modeling excercises. We also get to the point in some rendering sections that specific pages look terrible… etc and are going to need wireframes and better art production.

Claude, Grok, and Microsoft AI tools are greater than 95% accurate now with building simple web pages, especially using the predominate frameworks like Bootstrap.

6.3 Licenses.GreenvilleAssociates.com completely Rendered story board in itself.

7.0 STAGE WIREFRAME COMPLETE COMMENTS

This project as layed out is many orders of magnitude bigger than all existing commercial products in the same regard (See 3cx, cPanel who are Internet Leaders in this space) and is as large or equivalent to the project just completed in 547 as a Graduate student with teams as large as five.

Using Figma Design versus building pages is many times more complicated and slower than just building individual pages in your AI tool of choice. I am not sure it adds value. When you wireframe in Visual Studio you can make the screens work by dragging grid database objects on them.