

# DESIGN DOCUMENT VERSION2.0

FOR GREENVILLE LICENSE MANAGER (GEM.LM)



WIREFRAMES, DB ARCHITECTURE, BUSINESS LOGIC & OTHER DETAILS

## Contents

SECTION 1. EXECUTIVE OVERVIEW .....	4
1.1    Definitions of Wireframes.....	4
1.2    Figma Design .....	4
1.3    Figma Make.....	4
1.4    Greenville License Manager as part of Greenville Enterprise Manager .....	4
2.0 RENDER APIS IN PRODUCTION.....	5
2.1 Executive Summary.....	5
2.2 Porting And New Code from Azure and Google Cloud to Render .....	5
2.3 SCHEMA DESIGN FOR THE PROJECT .....	6
2.4 SWAGGER AND EXAMPLE OUTPUT FOR THE PROJECT – USERS.....	7
2.5 DIRECT ACCESS TO THE API FROM AN APPLICATION RETURNS A JSON .....	8
3.0 TARGET GREENVILLE LICENSE MANAGER WIREFRAMES – COMPLETE .....	9
3.1 Executive Overview – Translating Scope of Work to Target Designs. ....	9
3.2 Competitive Analysis.....	9
3.3 Significant Improvements over Competing Designs .....	10
3.4 Structure of Final Workflow includes Up to 18 Pages All of Which are Designed and in near working form.....	11
3.5 Menu Options DropDown & Pages Accessible via Top Nav Bar .....	11
3.6 LOGIN(Screen1).....	12
3.7 LOGOUT(Screen2) .....	12
3.8 User Notices(Screen3).....	12
3.9 User Help(Screen4) .....	13
3.10(Screen5) .....	13
3.11(6) Welcome(SystemInfo)(Screen6).....	14
3.12 Signup/Registration(Screen7) .....	14
3.13 All Licenses(Screen8) Legacy.....	15
3.14 Corporate Profile(Screen9) .....	16
3.15 ProductReviews(Screen10) .....	16
3.16 LicenseInformation(Screen11).....	17
3.17 Home2(Screen12) – Different View of Products – All Products in All conditions.....	17
3.18 LicenseLogs(13).....	18
3.19 Buy More(Screen14) .....	18

3.20 Upgrade Licenses(Screen15).....	19
3.21 Cartreview(Screen16) .....	19
3.22 Payment Review(Screen17) .....	20
3.23 Card Maintenance(Screen18) .....	20
3.24 Licenses.GreenvilleAssociates.com completely Rendered story board in itself.....	20

## 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 create 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 Anthropic's AI tool.

### 1.4 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.

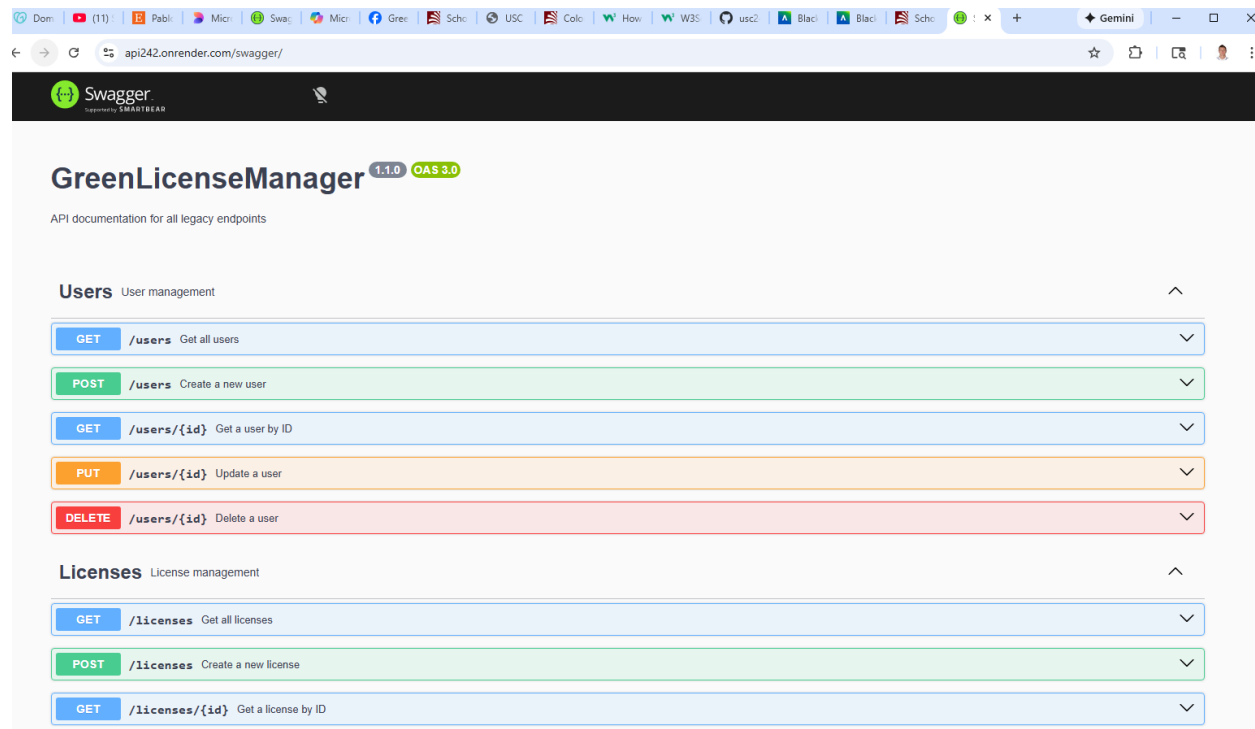
## 2.0 RENDER APIS IN PRODUCTION

### 2.1 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.

### 2.2 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.



## 2.3 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.

## 2.4 SWAGGER AND EXAMPLE OUTPUT FOR THE PROJECT – USERS

Swagger is returning the User Table Below

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://api242.onrender.com/users' \
  -H 'accept: application/json'
```

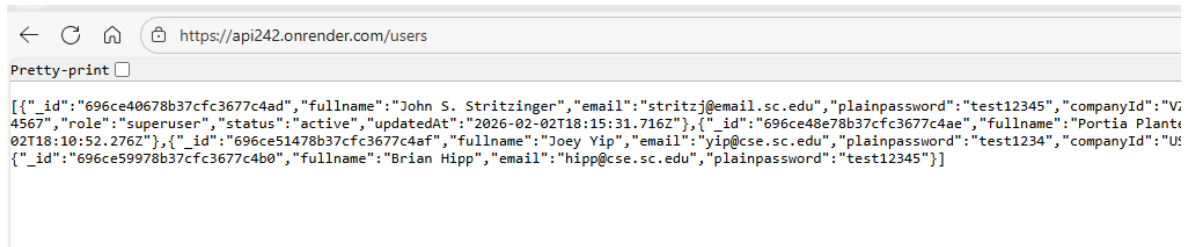
Request URL

```
https://api242.onrender.com/users
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "id": "696ce40678b37cfc3677c4ad",   "fullname": "John S. Stritzinger",   "email": "stritzj@email.sc.edu",   "plainpassword": "test12345",   "companyId": "VZFED0001",   "corporateuser": "True",   "password": "test1234",   "phone": "727-123-4567",   "role": "superuser",   "status": "active",   "updatedAt": "2026-02-02T18:15:31.716Z" }, {   "id": "696ce48c78b37cfc3677c4ae",   "fullname": "Portia Plante",   "email": "plante@cse.sc.edu",   "plainpassword": "test12345",   "companyId": "VZFED0001",   "phone": "803-867-5309",   "updatedAt": "2026-02-02T18:10:52.276Z" }, {   "id": "696ce51478b37cfc3677c4af",   "fullname": "Joey Yip",   "email": "yip@cse.sc.edu",   "plainpassword": "test1234",   "updatedAt": "2026-02-02T18:10:52.276Z" } ]</pre> <p>Response headers</p> <pre>access-control-allow-origin: * alt-svc: h3=':443'; ma=86400</pre>

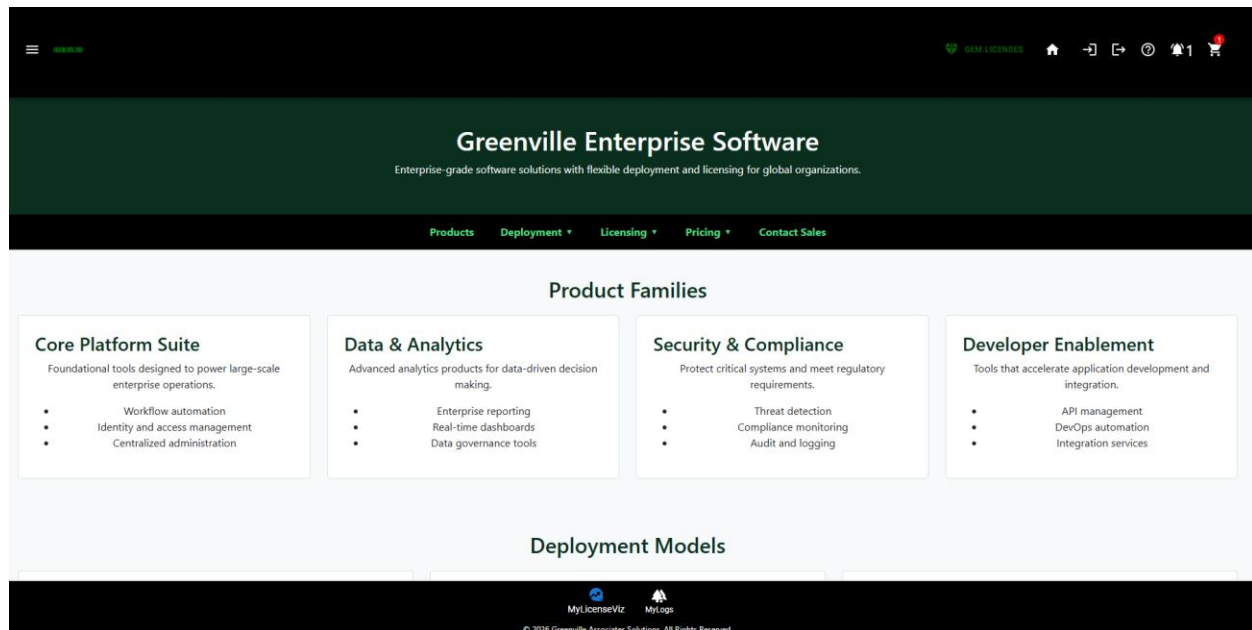
## 2.5 DIRECT ACCESS TO THE API FROM AN APPLICATION RETURNS A JSON



```
← ↻ 🏠 📄 https://api242.onrender.com/users
Pretty-print ☐
[{"_id":"696ce40678b37cfc3677c4ad","fullname":"John S. Stritzinger","email":"stritzj@email.sc.edu","plainpassword":"test12345","companyId":"V:4567","role":"superuser","status":"active","updatedAt":"2026-02-02T18:15:31.716Z"},{"_id":"696ce48e78b37cfc3677c4ae","fullname":"Portia Plant","updatedAt":"2026-02-02T18:10:52.276Z"},{"_id":"696ce51478b37cfc3677c4af","fullname":"Joey Yip","email":"yip@cse.sc.edu","plainpassword":"test1234","companyId":"U:1234"},{"_id":"696ce59978b37cfc3677c4b0","fullname":"Brian Hipp","email":"hipp@cse.sc.edu","plainpassword":"test12345"}]
```



## 3.0 TARGET GREENVILLE LICENSE MANAGER WIREFRAMES – COMPLETE



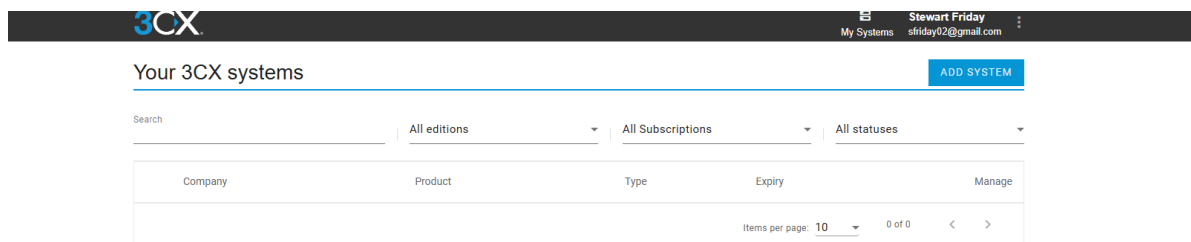
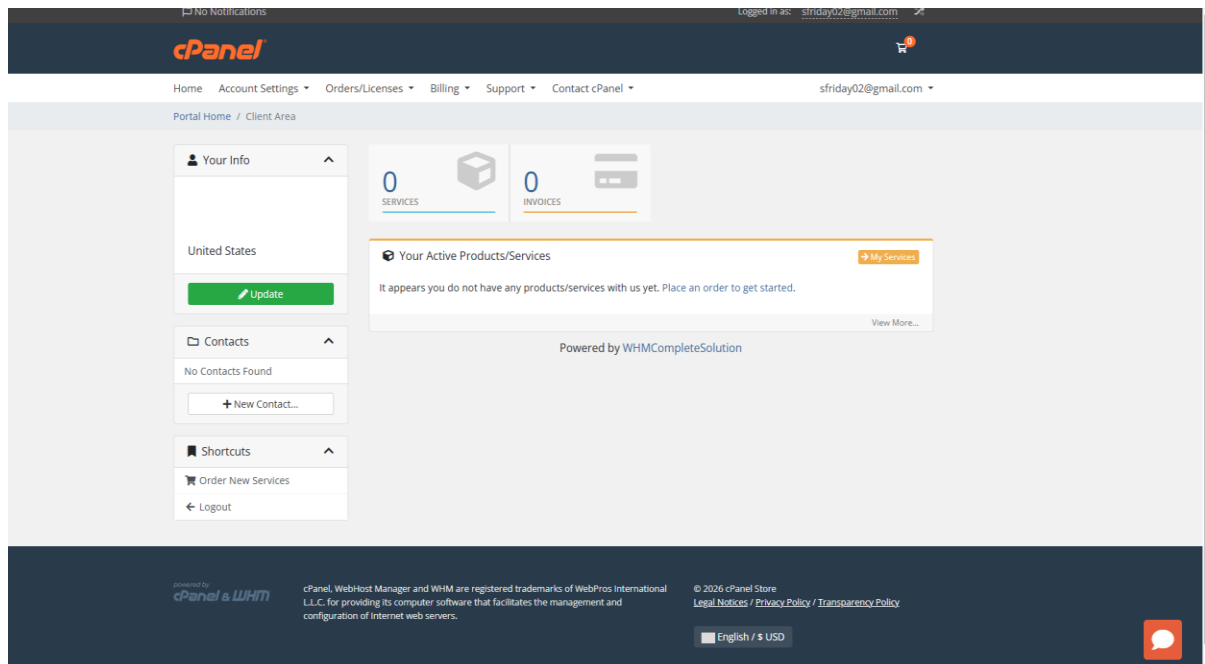
### 3.1 Executive Overview – Translating Scope of Work to Target Designs.

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

### 3.2 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:

- Our competitors tool only allows us to view our current licenses, add more and upgrade existing ones. This is significant code.
- 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.



### 3.3 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.

### 3.4 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.

### 3.5 Menu Options DropDown & Pages Accessible via Top Nav Bar

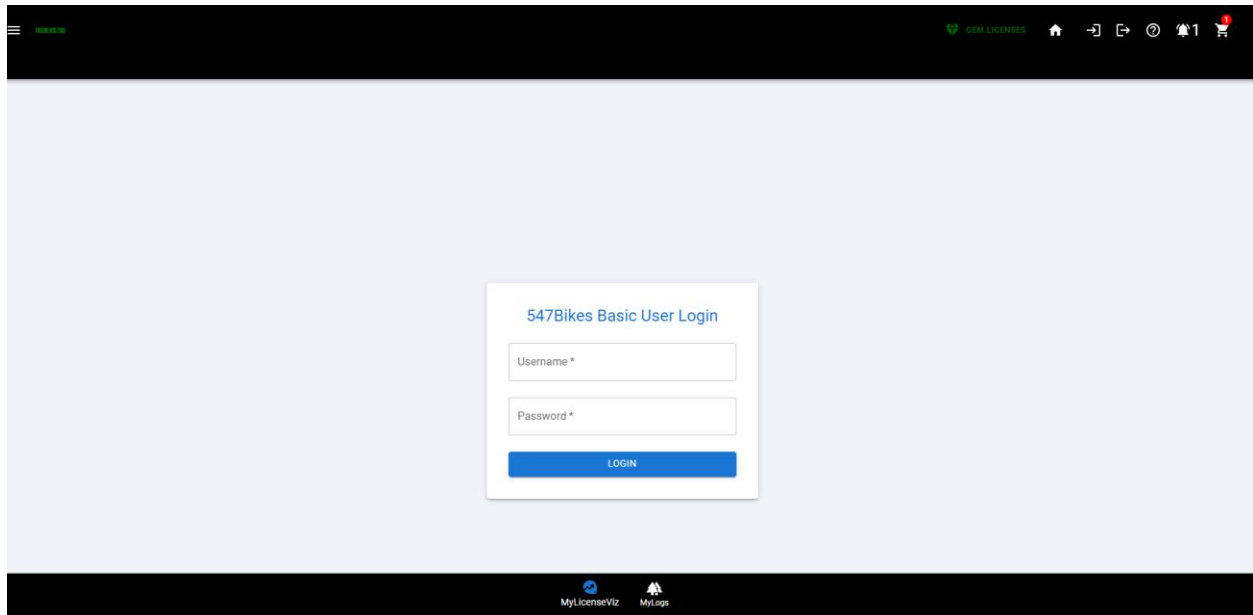
We intend to load the Sales Site as our Landing Page for the License Page with a General overview of the License Types available and where they can be used. Furthermore, the Login(1), Logout(2), UserNotices(3), UserHelp(4), And Cart(5) can be accessible from the Main Nav Bar and sized to work on all 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.



### 3.6 LOGIN(Screen1)

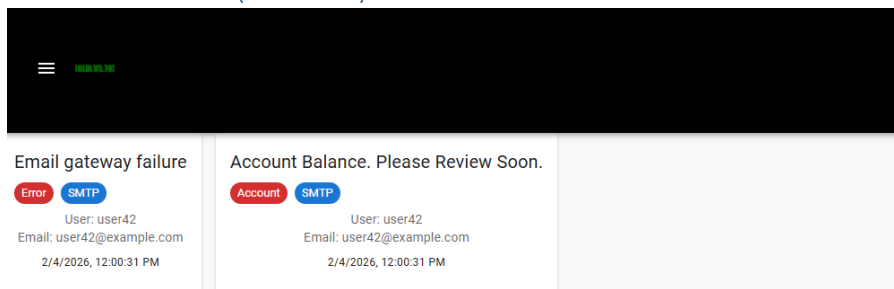


The screenshot shows a web application interface for '547Bikes Basic User Login'. At the top, there is a dark navigation bar with a hamburger menu icon, the text '547Bikes', and several utility icons including a heart, '547 LICENSES', a home icon, a share icon, a refresh icon, a clock, a bell with '1' notification, and a shopping cart icon. The main content area is light blue and contains a white login form. The form has a title '547Bikes Basic User Login', two input fields labeled 'Username \*' and 'Password \*', and a blue 'LOGIN' button. At the bottom of the screen, there is a dark footer bar with two icons and labels: 'MyLicenseViz' and 'MyLogs'.

### 3.7 LOGOUT(Screen2)

Processed in the background with a spinner.

### 3.8 User Notices(Screen3)



The screenshot displays a 'User Notices' screen. It features a dark header bar with a hamburger menu icon and the text '547Bikes'. Below the header, there are two notification cards. The first card, titled 'Email gateway failure', has a red 'Error' tag and a blue 'SMTP' tag. It contains the text 'User: user42', 'Email: user42@example.com', and a timestamp '2/4/2026, 12:00:31 PM'. The second card, titled 'Account Balance. Please Review Soon.', has a red 'Account' tag and a blue 'SMTP' tag. It also contains the text 'User: user42', 'Email: user42@example.com', and a timestamp '2/4/2026, 12:00:31 PM'. The background of the screen is light blue.

### 3.9 User Help(Screen4)

The screenshot shows a mobile application interface with a dark header bar. On the left is a hamburger menu icon and the text "GEM.LicenseManager". On the right are icons for a heart, a home button, a share icon, a refresh icon, a clock, a notification bell with "1", and a user profile icon. A modal window titled "Customer Contact" with a close button (X) is centered. It contains ten text input fields labeled: Phone, Email, Address 1, Address 2, City, State, Zip, Country, and Fax. A green "Submit" button is at the bottom of the modal. In the background, a button labeled "Submit Trouble Ticket" is visible. At the very bottom of the screen, a small copyright notice reads: "© 2026 Greenville Associates Solutions, All Rights Reserved."

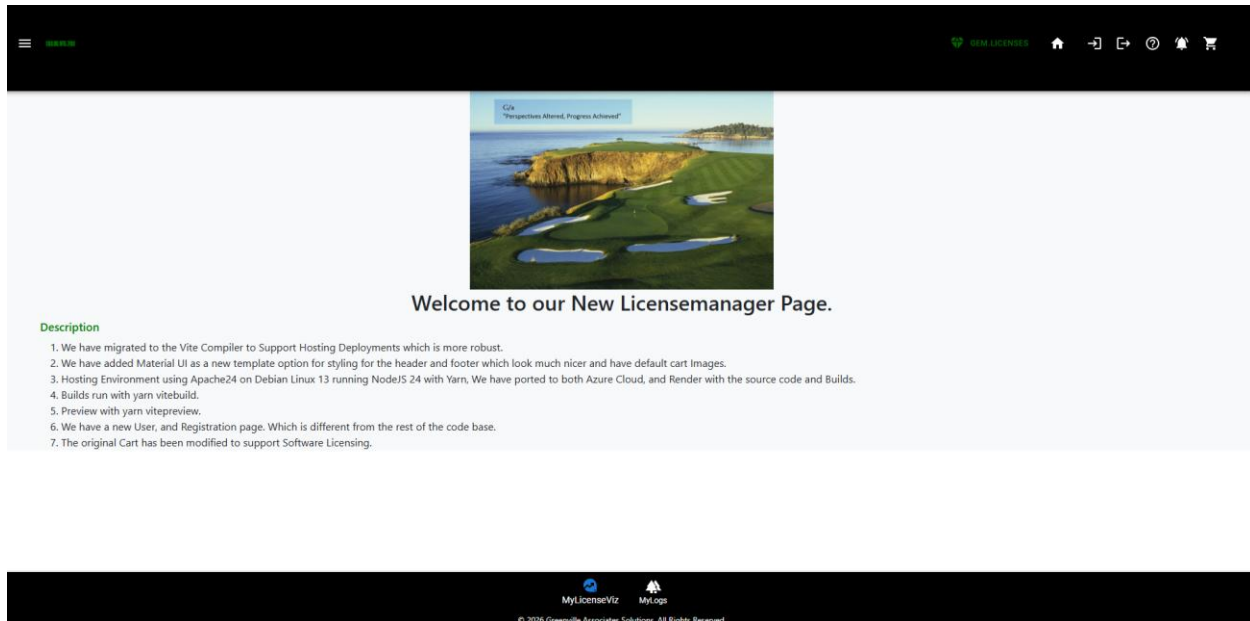
### 3.10(Screen5)

The screenshot shows the "Shopping Cart" modal in the GEM.LicenseManager application. The modal has a title bar with a shopping cart icon, the text "Shopping Cart", and a close button (X). It contains a list of items. The first item is "10CAL - FusionShell Professional V6.01" with SKU "FSPROV601-10CAL" and Vendor "Greenville Associates". It shows a "VP Discount (10%)" and the price "\$179.99" in green, with the original price "\$199.99" crossed out. To the right of the item is a quantity selector showing "1". Below the item list is a summary section with a table:

Subtotal:	\$179.99
Tax (8%):	\$14.40
<b>Total:</b>	<b>\$194.39</b>

At the bottom of the modal are two buttons: "Clear Cart" and "Proceed to Checkout".

### 3.11(6) Welcome(SystemInfo)(Screen6)



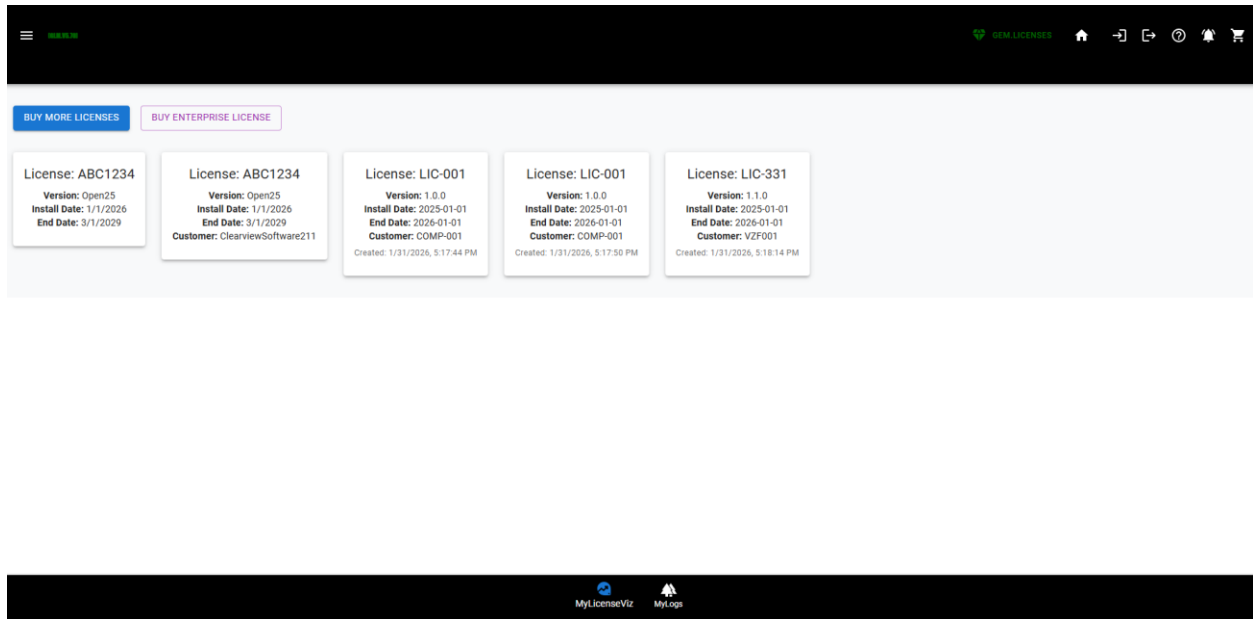
### 3.12 Signup/Registration(Screen7)

The screenshot shows a web application interface with a registration form. The form is titled "Register" in blue text. It contains five input fields, each with an asterisk indicating it is required:

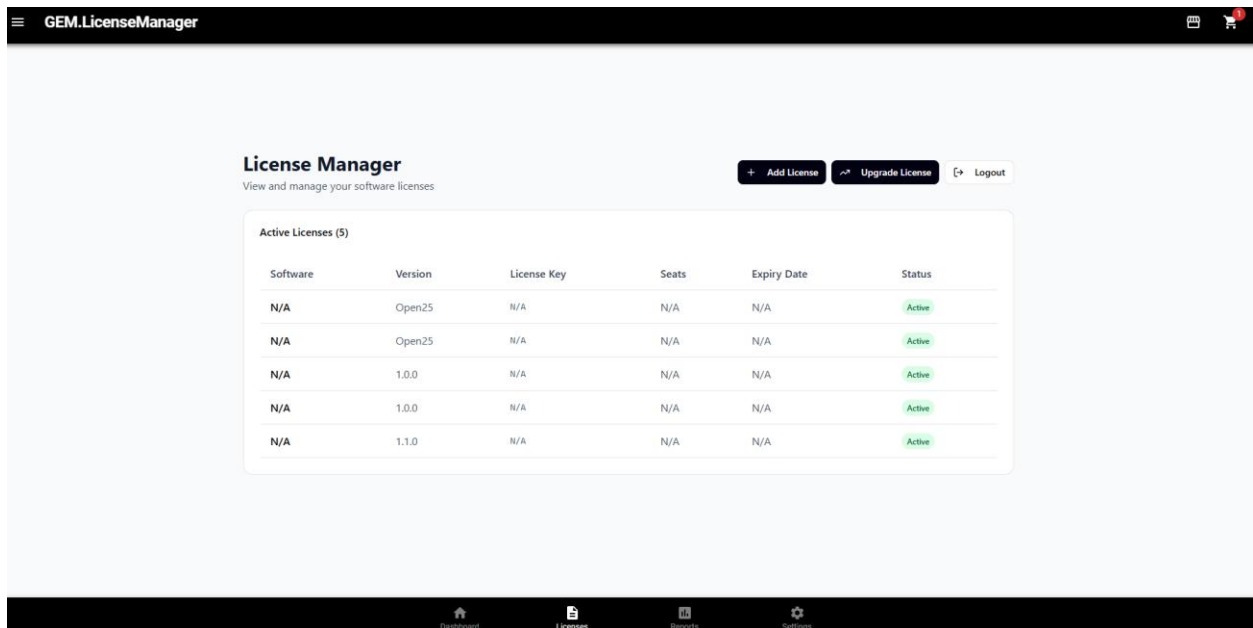
- First Name \*
- Last Name \*
- Username \*
- Email \*
- Password \*

Below the input fields is a blue button labeled "REGISTER". The form is centered on a light blue background. The top navigation bar and footer are identical to the previous screenshot.

### 3.13 All Licenses(Screen8) Legacy

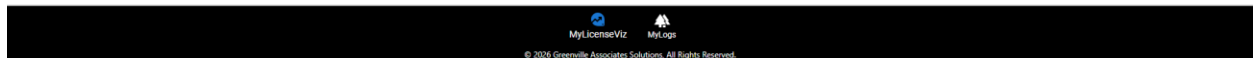
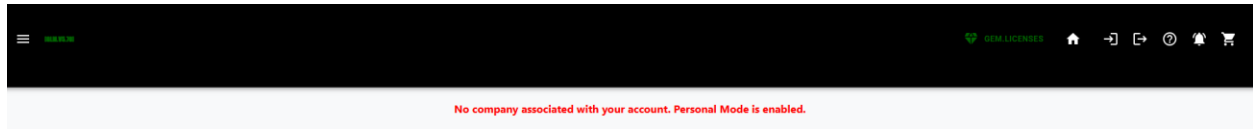


### Legacy Licenses – Figma Design(Improvement)

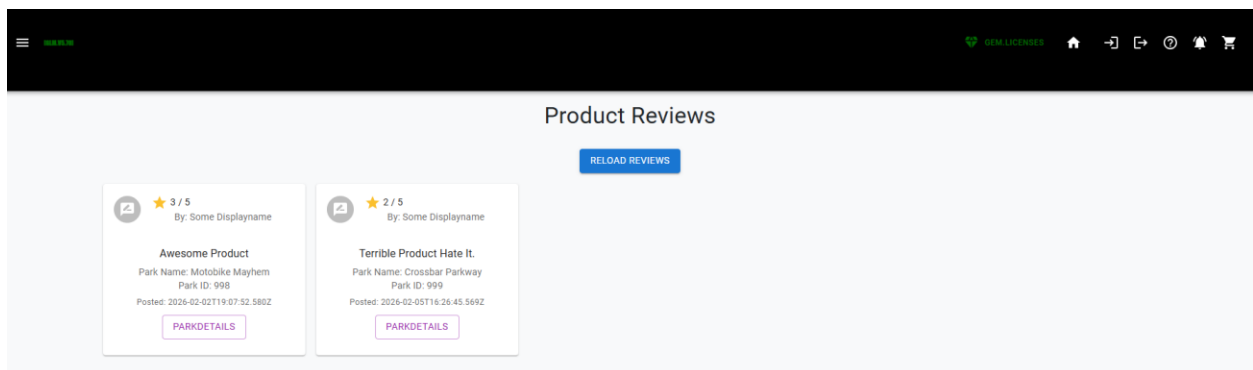


### 3.14 Corporate Profile(Screen9)

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.

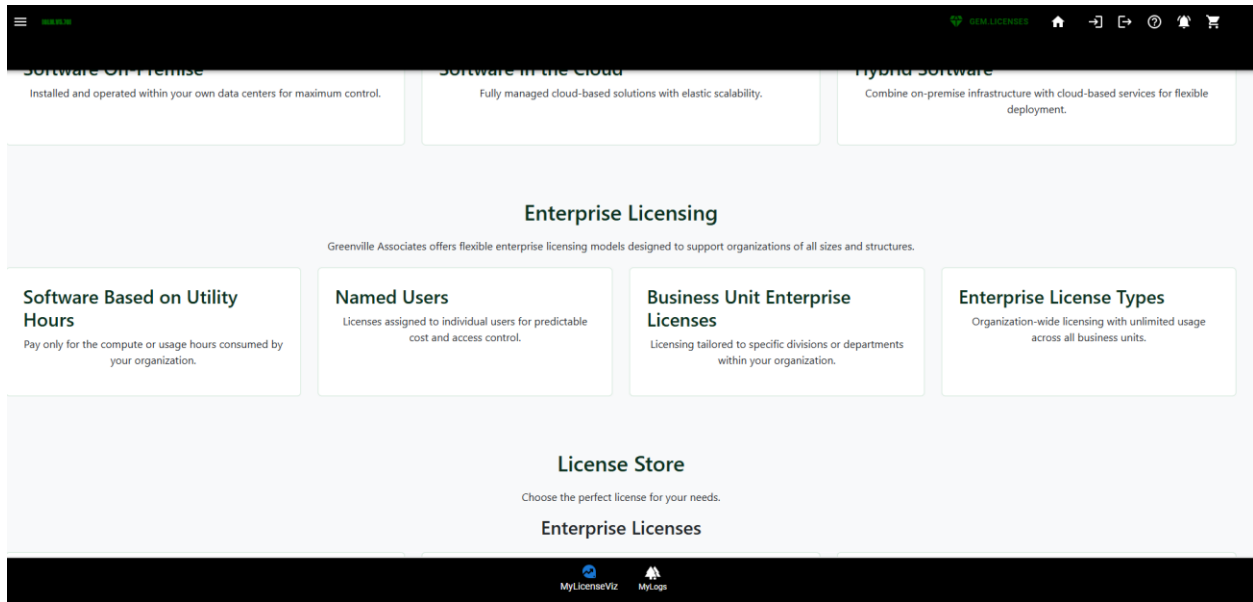


### 3.15 ProductReviews(Screen10)



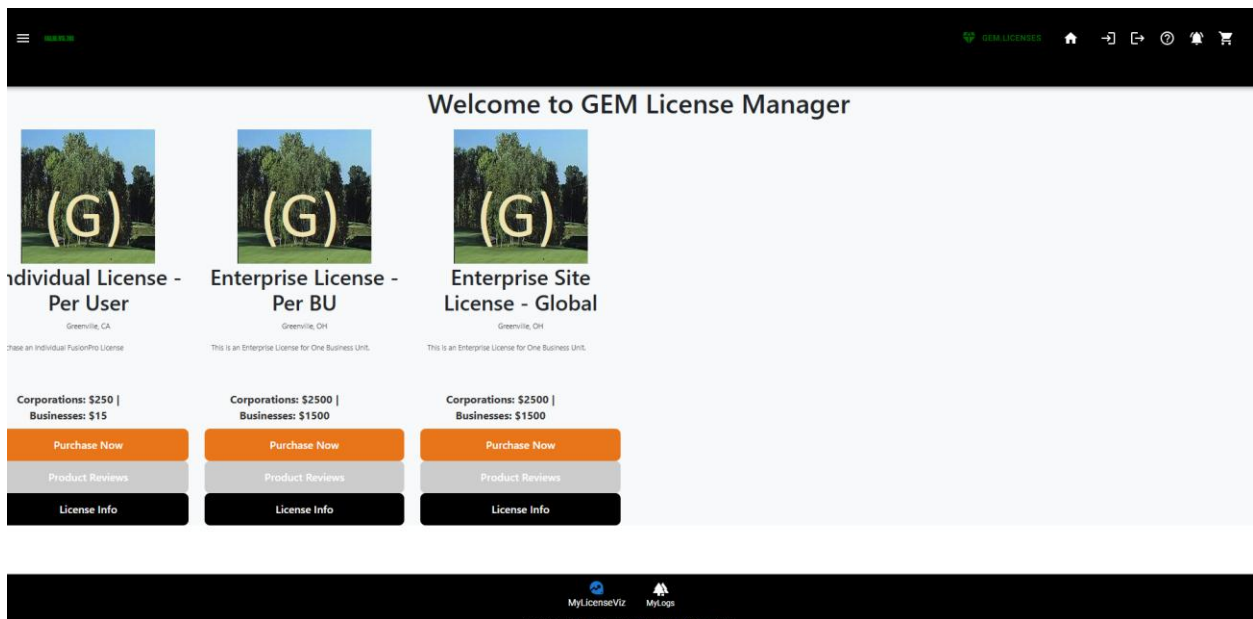


### 3.16 LicenseInformation(Screen11)

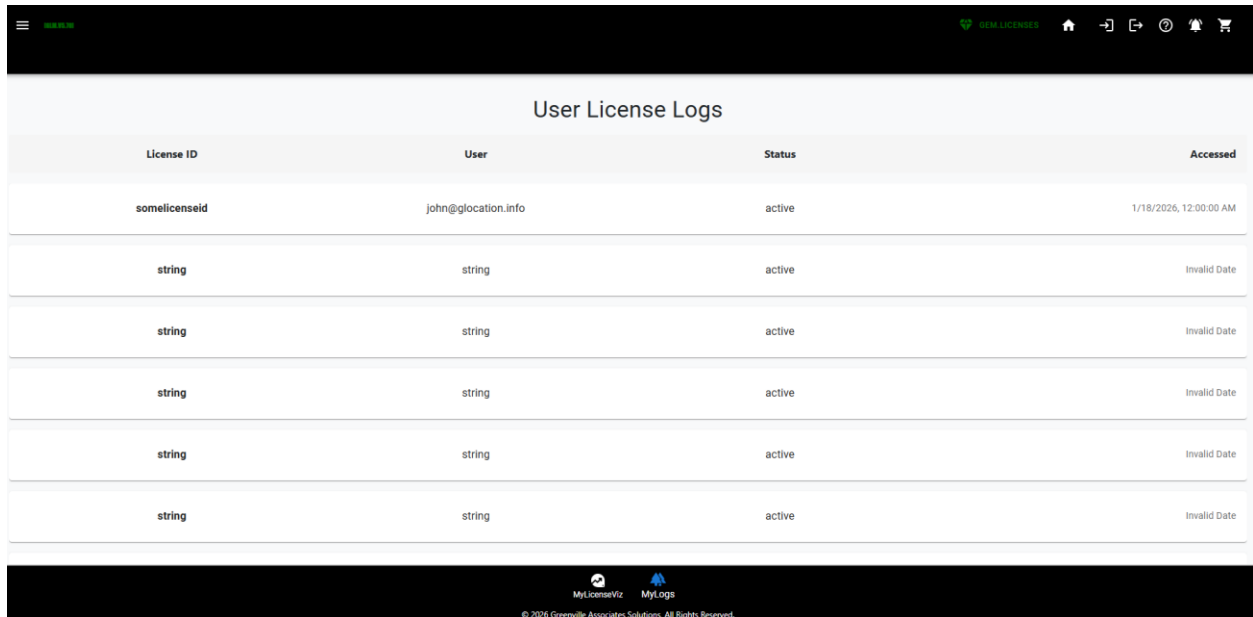


### 3.17 Home2(Screen12) – Different View of Products – All Products in All conditions.

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



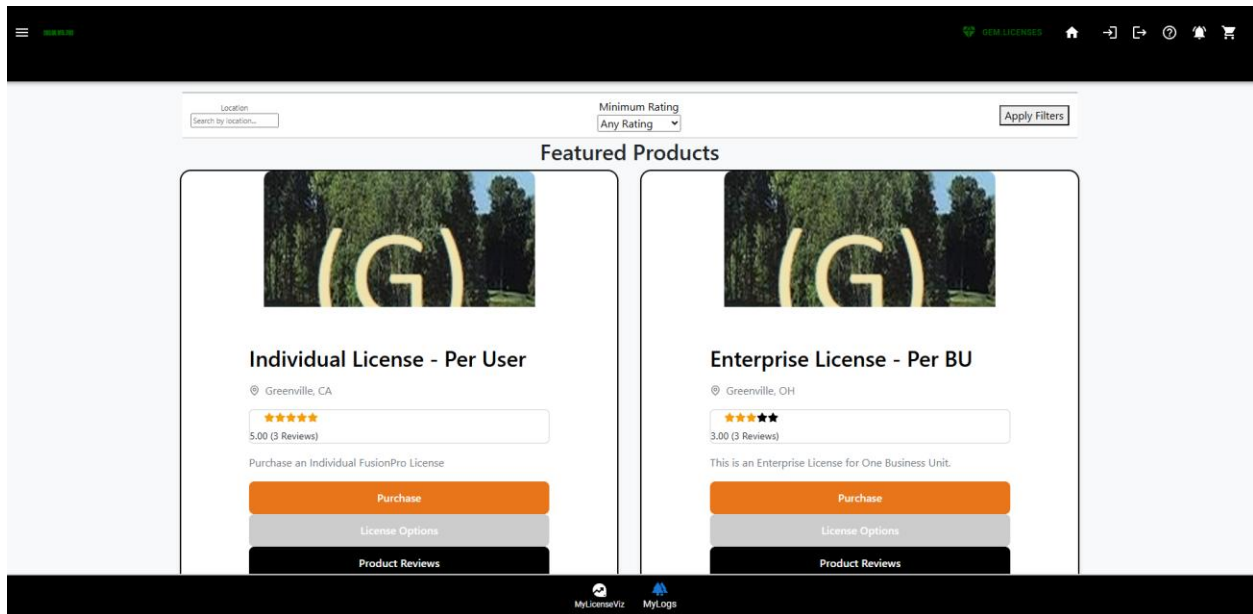
### 3.18 LicenseLogs(13).



The screenshot shows a web application interface for 'LicenseLogs'. At the top, there is a dark navigation bar with a menu icon, a 'MEM. LICENSES' link, and several utility icons. Below the navigation bar, the page title 'User License Logs' is centered. The main content area contains a table with four columns: 'License ID', 'User', 'Status', and 'Accessed'. The table has seven rows of data. The first row shows a specific license ID and user email, while the subsequent rows use placeholder text like 'string' and 'Invalid Date'. At the bottom of the page, there is a footer with logos for 'MyLicenseViz' and 'MyLogs', and a copyright notice for '© 2016 Greenville Associates Solutions, All Rights Reserved'.

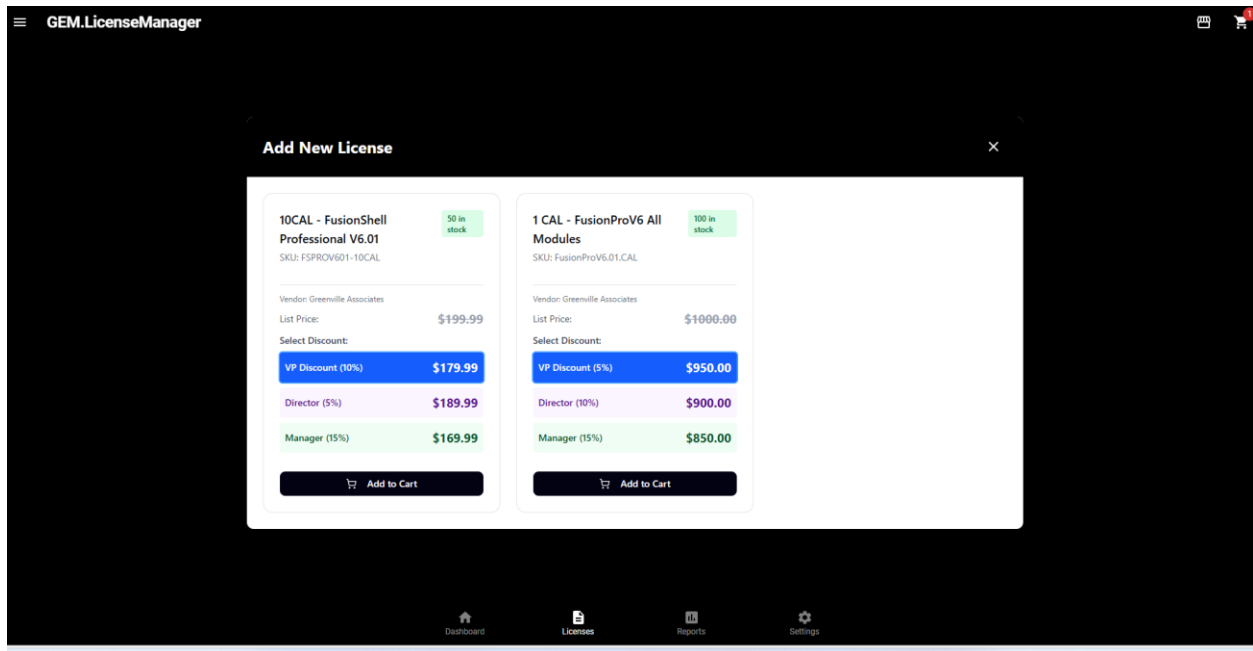
License ID	User	Status	Accessed
somelicenseid	john@glocation.info	active	1/18/2026, 12:00:00 AM
string	string	active	Invalid Date
string	string	active	Invalid Date
string	string	active	Invalid Date
string	string	active	Invalid Date
string	string	active	Invalid Date
string	string	active	Invalid Date

### 3.19 Buy More(Screen14)



The screenshot displays a 'Buy More' page within the same web application. The top navigation bar is identical to the previous screen. Below the navigation bar, there is a search bar with a 'Location' label and a 'Search by location...' placeholder. To the right of the search bar is a 'Minimum Rating' dropdown menu set to 'Any Rating', and an 'Apply Filters' button. The main content area is titled 'Featured Products' and contains two product cards. The left card is for an 'Individual License - Per User' for 'Greenville, CA', showing a 5.00 rating (3 reviews) and a 'Purchase' button. The right card is for an 'Enterprise License - Per BU' for 'Greenville, OH', showing a 3.00 rating (3 reviews) and a 'Purchase' button. Both cards also have 'License Options' and 'Product Reviews' links at the bottom. The footer is the same as the previous screen.

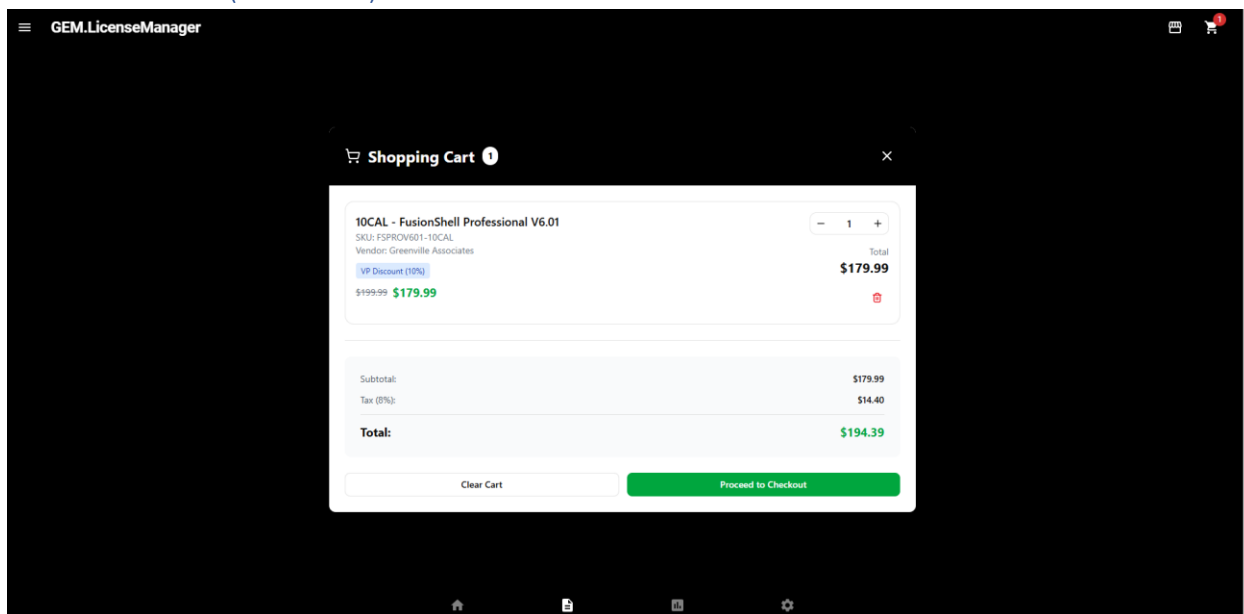
Inside the License Manager



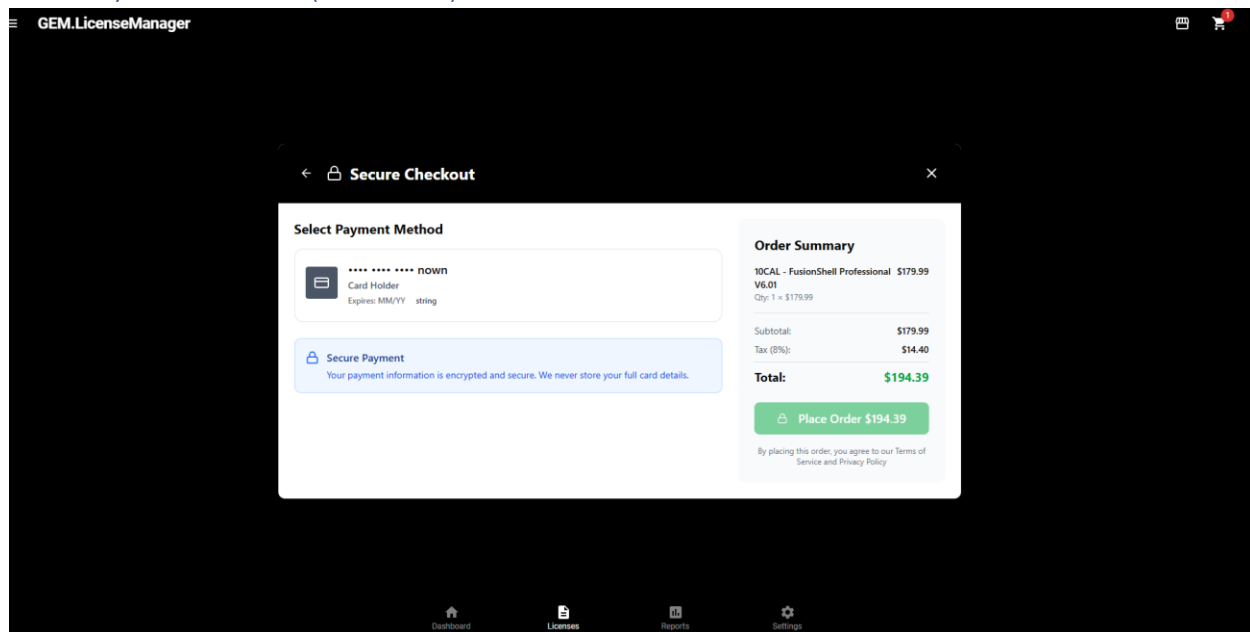
### 3.20 Upgrade Licenses(Screen15)

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

### 3.21 Cartreview(Screen16)



### 3.22 Payment Review(Screen17)



### 3.23 Card Maintenance(Screen18)

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.

### 3.24 Licenses.GreenvilleAssociates.com completely Rendered story board in itself.

We have built a website which more or less already expresses our wireframes in HTML. We also have a ReactEndUser Site, and ReactAdmin site which is in excess of this. In other words we have three working prototypes against one set of wireframes, and a 33 table database supporting them.