John S. Stritzinger
Project Proposal for CSCE242

PROPOSAL REQUIREMENTS

a) The name of your website
b) The description of your topic: must contain at least 5 sentences
c) A list of each of the page names
d) A description of the data you will be storing, manipulating and viewing.

PROJECT PROPOSAL PART1

Proposal Option1 – MyLinkV3.0

a) mylink.glocation.info

b) ENTERPRISE HR MODULE FOR RESUMES REPOSITORIES

Mylink is intended to be a generic competitor to LinkedId offering the ability to build a vitae with work history(i), education(ii), activities(iii), certificates(iv), published works(v) and interests(vi). The difference is we view resume building as part of an extended enterprise HR application where users are storing resumes for the enterprise HR use case, and not necessarily a standalone website for residential users. Mylink components should exist as a standalone app, but should also be generic components which can be used in the Fusion Product family as necessary.

I previously took CSCE590 as a Graduate Student with Dr. Achanti, and we worked on a ISP User Certification Database where we built Azure scripts to track Certificates earned by Employees, and ISP users alike.

I have been working on an extension of this platform/rebuild which would allow someone to enter an entire resume including certificates. To do this I had already rebuilt all the code for the class project and have ported a new auth structure to SQLITE. I would like to take this very large tool and make it customer ready/investment ready as it's a good user of time. The name of the Application is currently codenamed MyLink which is a variation of LinkedIn and has most of the same features.

MyLink however was initially called CertAthalon, but we are intended to manager Enterprise Level Information which is a different task. We need to track how many resumes we have stored, etc. At some companies I have worked, storing your resume was a requirement to employment.

c) **Pages** include Login, Logout, UserProfile, Admin/Manager, User Resume View which is a complicated CRUD front-end for six tables, Enterprise Reporting on Stored Resumes, QueryEngine which would allow you to sort for specific keywords, and Business Unit Reporting (More than 3 reports each of which is a page). This is a very large app, which is built in part all in Javascript. MyLinkV2 is using a C# API set and HTML only. So this is a major port/rewrite to Express on a different database with more than five new pages, and at least six ported pages which we can identify during the build.

d) Data Stored: User Personal Details in form of Online Vitae for unlimited numbers of users, user photos, and CRUD operations for the six main categories listed above. We also need to build Enterprise Auth capabilities as the previous app was for ISP users only.

Proposal Option2 – FusionShell License Manager

a) **licenses.greenvilleassociates.com**

b) **LICENSEMANAGER FOR OPENSOURCE SOFTWARE**

As you have seen before, my company Greenville Associates has an Opensource Software Solution called FusionShell. It has a default form layout and sample pages which currently are not connected to a post architecture. The solution runs on PHP now, and you saw a form of this in my research project.

LICENSEMANAGER PORTAL A NEW CONSTRUCTION AND PERMAMENT EXTRANET.WEB COMPONENT

What I would like to do for my project as an alternative is to build a licensemanager for the Installer so that when people download and install my tools, I get a license built on a public website. So this would be just for the Licensemanager which has a local component, and Network Utility for existing customers where they can view their licenses, and their subscription status, and buy more software, or upgrade software.This is an important tool which I need to build anyway. The system creates a 30 day license now locally but it doesn't track it over the network. Its locally annoying after 30 days, assuming people would call to fix it.

OPENSOURCE USAGE TRACKER

Along with the licensemanager, I also want to track usage of my product, and that would be to allow a log to be posted on each usage. This would send the license id and the date of access to the controller.

CART-PURCHASE UPGRADES

Once the licensemanager is reached I need the ability to buy a new license and this requires a cart system which I have written in part from my 547G class this fall but was not graded. I am proposing including the FusionCart with this bundle and it should be styled with roughly the same capabilities of the 3CX Voice over IP Manager but using a different form obviously and different purpose.

c) **Pages** include LIcenseManager Home with Marketing, LicenseManager Login, and Logout(requires Auth), License View, License Upgrade, BuyNew, Cart & Intermediate Pages(5), Add CreditCard/Payment Type, Contact Support/Open A Ticket, SecurityManager.

d) **Data Stored:** Users, Licenses, Access Logs, Heartbeat Data from the App, Shop Options/Cart Product Catalogue, CartMaster, Cart, CartItems > 10 tables likely.

Comments:

The first option is a bit complicated which Ms. Plante has seen in part as Auth exists in some form for what our team called the GhostPortal. However that wont work on Express so it would have to be rebuilt to the same standard using a different DB. I have a layout for the pages, but they don't support CRUD, and this would be an entirely new backoffice.

The second option is an entirely new build from the ground up but is a bit lower priority(although high priority) for my company Greenville Associates.

Since I have to complete my graduate research which is a very complex project I am hoping to get the project done early in the term rather than later in the semester. I have my endpoints working as of this morning 1/18/2026.

Thanks,
John