**Project Overview**

Based on the challenges described by the principle, teachers, and IT staff with students who are distracted by social media, along with the potential for students to use AI to do their classwork, we propose developing a specialized web browser to be installed in all student computer labs, the library workstations, and computer classrooms. This browser would allow for teachers to place restrictions on which websites a student is prevented from visiting. The browser will also provide students with a list of bookmarks to sanctioned academic materials. Lastly, deny lists and bookmarks will be centrally managed, allowing teachers to publish updates from any instance of this browser, without the need to burden their limited IT staff.

We envision this project to be delivered in three phases, taking roughly two weeks each. Phase one will include the development of basic browsing functions and scaffolding needed to support future phases. Phase two will include the ability to block specified websites and for teachers to be able to maintain the block list. Phase three will include the ability for teachers to publish bookmarks to sanctioned academic resources.

**Vision Statement**

Empowering educators with a secure and controlled web browsing environment for effective classroom management.

**Elevator Pitch**

See file “CS6920 Elevator Pitch and Ad.png” in the same directory as this document.

That includes a text description and image that would be used for a print advertisement.

Recommend right click and “Open Folder in Fire Explorer” as Visual Studio default viewers don’t work well on MSWord and other non-code files.

**Business Problem**

We need to provide our K12 students with computers and the ability to access academic resources on the web. The challenge is that our students often get distracted surfing social media, playing web games and the web in general, rather than focusing on the academic resources they should be using to complete their schoolwork. They also may use various unauthorized resources to look up answers to test questions or AI to do their assignments. We would therefore like to replace the browser with one that allows for more control over what sites students are allowed to visit.

**Identified Features for Project**

1. A working basic web browser.
2. Ability to limit what websites students can visit.
3. Ability to provide bookmarks to sanctioned educational websites.
4. Ability to central manage configuration

**Feature 1 User Stories – A working web browser**

1. As the principal of this K12 school, I am facing declining scores on standard benchmark tests which endangers our federal funding. I need a solution to all of these online distractions so that we can get things back on track.
2. As a teacher I need students to be able to do online tests and view academic web resources.
   1. What features do you need this browser to support? For instance, does it need a URL bar and forward and back buttons.
      1. Yes, and a reload button.
   2. Are there any additional features or controls that you think students will need?
      1. I need them to be able to open multiple websites at once.
   3. So, would a tabbed browser be what you are looking for?
      1. Yes.

**Feature 2 User Stories – Deny access to inappropriate Websites**

1. As a teacher, I want to limit a student’s access to certain distracting websites, so they stay on task.
   1. Do you want us to block all websites except those that are allowed, or to allow all sites except those that are blocked?
      1. I just want to block students from going to sites, such as Facebook, Instagram, Twitter, Tiktok, etc.
2. As a teacher, I want to prevent students from utilizing AI websites to cheat.
   1. What additional sites would need to be blocked?
      1. The big problems we see are with ChatGPT and Co-pilot, but there may be others.
   2. Do you need a way to update the list of blocked sites.
      1. Yes, that would be great.
   3. When a student visits a blocked site, what should happen?
      1. Can we send them a message and tell them to get back on task.

**Feature 3 User Stories – School provided bookmarks**

1. As a teacher, I would like to be able to provide the students with a preset list of bookmarks that point to approved academic resources.

**Feature 4 User Stories – Central management**

1. As IT support for the school, I would like for teachers to be able to maintain a central blocklist from any computer, so that I don’t have to spend my time updating block lists all day.
2. As IT support for the school, I would like the same for bookmarks for the same reasons.

**Goals for Iteration 1**

We will implement both User Stories in Feature 1 and prep infrastructure and proofs of concept needed to support User Stories in Feature 4. The visible deliverable to the customer will be a basic working browser, but we will also establish the infrastructure for hosting a central configuration service using a REST API. This will include the hosting of the service, and a test API to prove our ability to store and retrieve data from the remote service.

Work breakdown for this will be in github issue tracking database tagged as “tasks” and set for milestone “Iteration 1”. Tasks done during this phase are closed and marked with milestone “Iteration 0”.

List of tasks from issues log included here for convenience:

Jim

Feature 1 – Menu bar

Feature 1 – Tool (button) bar

Feature 1 – Working Forward, Back, Reload Button

Feature 1 – Working Create Tab button

Feature 1 – Working URL bar (to go to a site)

Feature 4 – GUI Placeholder for “Teacher Mode”

Feature 1 – Working Browser window

Feature 1 – Tab bar with close buttons

Technical – Set up project for Unit testing

Technical – Add JSON Library

Feature 4 – Add Create test object model (group)

Davide

Feature 4 – Create Object Serializer

Feature 4 – Create Object Deserializer

Feature 4 – Create REST get call (Read)

Feature 4 – Create REST post call (Create)

Feature 4 – Create REST patch call (Modify)

Feature 4 – Create REST delete call (Delete)

Feature 4 – Create DAL to abstract remote object management

Feature 4 – Secure Hosting service(s)

Feature 4 – Configure Hosting service to support Python Flask

Feature 4 – Setup MySQL backend for use by Python Flask services

Rob

Feature 4 – Setup Flask

Feature 4 – Load additional Modules required for REST API

Feature 4 – Load additional Modules required for MySQL communication

Feature 4 – Create MySQL database table(s) for test Object

Feature 4 – REST endpoint to read JSON object

Feature 4 – REST endpoint to create JSON object

Feature 4 - REST endpoint to modify JSON object

Feature 4 - REST endpoint to delete JSON object

Feature 4 - Testing framework for python