Capstone Project

Daniel Dickson

School of Business, Liberty University

CSIS 484: Information Technology Capstone

Dr. Daniel O’Malley

May 5, 2022

Table of Contents

Abstract3

Introduction4

Conclusion Report5

Biblical Worldview6

Project Closing7

Lessons Learned Report8

Scope, Time, and Cost Goals8

Success Criteria9

Lessons Learned10

Conclusion11

References12

Abstract

With the growing issue of data theft and cybercrime, many organizations have quickly recognized the need to implement some form of private networking. While in-person conversations can eliminate most of these problems, ideas conveyed in this manner are much more likely to be forgotten than they would be if they were documented onto a secure private network. Throughout this course, a rough mock-up for a private forum was developed to illustrate how developers can address this need. Although the system remains basic in appearance and functionality, in addition to the likely event of it being outperformed by services such as cloud computing, it has illustrated the need for developers to save information in a privatized manner. Throughout the curriculum, various guidelines concerning graphic design and IT have culminated to illustrate many of the basic needs that must be met in the field of IT, and user feedback received throughout the course have regularly helped to reinforce these concepts, even as the project officially enters its final stage of development.

Capstone Project

Throughout the entire major, classes in computer programming, graphic design, and data management have helped to develop the necessary skills for effective software development. Many textbooks concerning languages such as Java, Python, and SQL have illustrated many of the core programming principles required for effective systems development, allowing the successful implementation of a four-page web application before this class had started. In response to criticism from the instructor, a database relying on PHP and the XAMPP web server was also integrated into the system. Despite how none of the courses in this major had seemed to discuss these two services in any significant detail, numerous online resources have documented how these systems can potentially be implemented.

Communications, philosophy, and cultural studies classes have also displayed the need to remain aware of potential differences in cultural upbringing and overall worldview. Various textbooks have shown why developers should actively welcome these differences, with one business communications textbook saying, “a letter, memo, email, or phone call reflecting a ‘you attitude’ sends a direct signal of sincere concern for the receiver’s needs and interest” (Lehman, DuFrene, & Walker, 2017, Sec. 3-4a). The current textbook shows a similar sentiment, as it describes user feedback as a necessary aspect of systems development (Satzinger et al., 2016). This was done throughout the major, because differences in priorities and worldview can greatly influence how developers and end users may prefer to approach development and prioritize intended features, and the ability to recognize these differences when reviewing feedback can raise awareness of major issues that may otherwise be overlooked. Throughout this project’s life cycle, feedback from other students and the instructor has greatly influenced this project’s development to ensure that it can truly meet users’ needs.

**Conclusion Report**

In much of this class and in CSIS 474, this project was developed to address a major issue within the Valve Corporation, where many videogame developers’ concepts remained underappreciated due to groupthink and to lack of knowledge in how developers can share their ideas with colleagues and potential customers (Stewart et al., 2018). Thankfully, reports indicate that the organization seeks to address this issue and remain open to new ideas, even treating failed projects as an acceptable norm (Jha, 2020).

As this final assignment officially closes, various changes have been made to ensure the system will meet business standards of database management. Various upgrades in color scheme, window layout, and operations have ensured the system can fulfill its intended purpose and thoroughly represent how it would operate if it were integrated into an official work environment. Admittedly, the website’s appearance and functions remain largely basic despite now having a XAMPP database. In addition, there is a high chance that official organizations will have already installed some other medium of communication such as secure emailing or cloud computing, which could arguably fulfill a much wider range of needs. However, this system was only meant to raise awareness of the need for private networking and how the issue could potentially be addressed. Throughout this class, the system also relied heavily on the agile development approach, relying heavily on user feedback to represent techniques that official developers would use to recognize and address their clients’ needs. Even if this specific system is never adopted, consideration of biblical worldview, production goals, and methodology taken throughout development have proven to remain applicable across the field of IT, regardless of the methodology used to illustrate these considerations.

**Biblical Worldview**

Biblical principles have regularly shown the need to remain open to feedback and avoid working too hastily, as is also promoted in the textbook when it describes the alternative as “cowboy coding” (Satzinger et al., 2016). Scripture regularly contains verses warning, “Wealth gained hastily will dwindle, but whoever gathers little by little will increase it” (Proverbs 13:11, ESV), and elsewhere saying, “Desire without knowledge is not good, and whoever makes haste with his feet misses his way” (Proverbs 19:2, ESV). Principles such as these will be necessary as developers enter the workforce, as these doctrines will encourage them to plan their work more carefully and divide major projects into more feasible milestones.

Many other verses also emphasize the need for honesty in the workplace, saying, “Do not lie to one another, seeing that you have put off the old self with its practices” (Colossians 3:9, ESV). This consideration is especially significant, as its absence could allow developers to exaggerate their performance, or it could leave the organization unaware of major issues. These errors could eventually cause the organization to place too little effort to meet deadlines, lose sight of potential health problems such as seizures, or even neglect major security weaknesses, leading to incidents such as the infamous Sony hack (Kosseff, 2018). The result could be especially disastrous for systems that are necessary to meet basic needs such as heating or water filtering. In other cases, dishonesty could embolden developers to prioritize personal gain despite having promised to focus on users’ needs. This mentality often results in fraud, with developers ignoring established security principles such as separation of duties to use funding beyond the organization’s intended goals (Johnson, 2015), as is widely known to have happened at Enron (Eckhaus & Sheaffer, 2018). These considerations will be essential in the field of IT, especially as society becomes increasingly reliant on technology.

**Project Closing**

In this final phase, there was another attempt to integrate the SQL database into the HTML web application. After a brief email exchange with the instructor, he suggested that many of his students had used a program named XAMPP for this purpose, which was bundled with phpMyAdmin. The previous SQL file was initially used as a basis for this new system and has been kept for future reference.

As stated in the previous phase, the system is intended to allow testing on other devices, regardless of whether they can connect to the internet. As a result, the system only requires users to install the XAMPP program and place the attached files into the appropriate locations. Since the full XAMPP folder would have taken over half a gigabyte of hardware space, only some HTML files, their accompanying PHP files, and the “projectdb” folder have been attached for the sake of convenience. The HTML and PHP files are to be stored in “C:\xampp\htdocs,” and “projectdb” should be placed in “C:\xampp\mysql\data.”

As shown in last week’s discussion board thread, the system often returned a simple blank page on Firefox browsers despite having code in place to output errors. Recent tests on Firefox have returned random strings of code as well, and Chrome had simply returned the raw PHP script for unknown reasons, both of which could have led to major security issues if the database had contained actual users’ information. In previous tests, the page only succeeded to record one row of data in which every input box was filled with data. In the belief that the data could be replaced with this same method, it was promptly deleted. These risks illustrate one major reason why the system remains relatively simple in function, with no true connection to any official server or forum.

**Lessons Learned Report**

Despite any major problems that may still be present within this system, the code present within the system should effectively represent how the system is intended to operate. Resources concerning PHP have helped to provide a general idea concerning how this HTML system can be implemented with support for SQL data. Throughout this assignment, development has revealed various practices in scope management, success criteria, and general approach that can greatly enhance future endeavors. Screenshots in the appendix show how this system currently operates with this service.

***Scope, Time, and Cost Goals***

Since the instructor had called for this course’s project to be based on the ones described in CSIS 474, the system already had four HTML pages complete with CSS formatting, JavaScript prompts, and checks to ensure the correct data had been entered. The inclusion of these pages allowed the project to start development much more easily than if these pages were absent until the beginning of this project. Aside from this inclusion, the only scope, time, and cost goals were to ensure the project remained simple enough that it could be completed in a college setting without spending large sums of money on hardware or software. Submissions to my previous class did include database hardware, firewalls, and switches, most notably appearing in network diagrams, but these systems were only included to represent how the system would appear if it were integrated into an official organization. Since implementing these systems would have required direct contact with Valve’s personnel, in addition to acquiring enough funding to implement the service on an international scale, these components were promptly rejected for the purpose of this illustration.

Since each company can have its own set of goals, the system would also have likely needed to be heavily modified or discarded if it were proposed to a real organization. There is a high chance that this organization would already have selected a more effective means of solving the same problems that this proposed system was meant to address. As a result, it would not have been worthwhile to implement the system on an international scale, even assuming the necessary time, resources, and personnel were available.

***Success Criteria***

Naturally, the project needed to be significantly downscaled from its original version, so it could be feasible within the scope of this illustration. Aside from the main goal to provide a general illustration of the system promoted in CSIS 474, the only success criteria in this assignment were to address any criticisms received by other users and the instructor, due to the project’s heavy emphasis on agile development. The system was never meant to be a complete forum, as it would have required a stable internet connection and either local server hardware or cloud computing to maintain. While some viewers may consider this decision to be a missed opportunity, others may believe this compromise to be necessary to ensure that one does not spend too much time or resources on a system that may need to be modified or rejected. Decisions such as this were intended to help prepare for agile development, which often requires substantial shifts in software requirements as business needs can abruptly change.

***Lessons Learned***

Projects such as this have helped to show the error in overthinking how one should approach problems, as developers can become so reliant on user feedback that they may underperform if user feedback takes an abnormally long time to arrive. This news should not in any way be treated as an insult to users or the feedback they provide, as they may experience complications such as poor internet connection, heavy workloads, or a DDoS attack. Instead, developers should frequently combine user feedback with their own benchmarking, dismissing any personal bias to recognize both positive and negative aspects of their work.

Overthinking has led to the consideration of unnecessary features that would have caused the project to become too ambitious. For example, proposed cosmetic upgrades such as the top navigation bar and background image were discarded to better reflect how actual web forms would appear and to avoid seeming unprofessional. While it is essential for developers to consider users’ feedback and adapt to meet those users’ needs, it is also important to understand how to solve problems independently until the necessary feedback arrives.

Arguably the main lesson learned from this assignment is to bring multiple ideas for projects into consideration, determine how each system will need to be implemented, then decide which implementation would be the most feasible. Before development even begins, this approach can raise awareness of strange behaviors that may occur during development, including the persistent white screen that plagued this project despite having code in place to output errors. Developers can then eliminate many prominent errors much more effectively and prevent others from ever occurring.

**Conclusion**

After many weeks of production, the project should illustrate the general need to establish a private means of storing important data. Most other shortcomings previously present within this project have been addressed throughout the past several weeks, especially with the textbook recommending focus on only a few tasks at a time (Satzinger et al., 2016). Thorough information from previous courses and from colleagues have also identified various weaknesses that must be addressed to allow effective systems development. Personal benchmarking, biblical worldview, and several lessons learned in this course have further improved performance in this project and provided ways to prevent any major errors present in this project from appearing in future endeavors. Each of these considerations can greatly improve efficiency and overall performance in nearly any project in the field of IT. It has been an honor to see these effects firsthand throughout the IT curriculum.

References

Eckhaus, E. & Sheaffer, Z. (2018). Managerial hubris detection: The case of Enron. *Risk Management, 20*(4), 304-325. <http://dx.doi.org/10.1057/s41283-018-0037-0>

Jha, S. (2020). Organization without hierarchy: *A case study of Valve Corporation. Review of Management, 10*(1/2), 19-24. <http://ezproxy.liberty.edu/login?qurl=https%3A%2F%2Fwww.proquest.com%2Fscholarly-journals%2Forganization-without-hierarchy-case-study-valve%2Fdocview%2F2579426552%2Fse-2%3Faccountid%3D12085>

Johnson, R. (2015). *Security policies and implementation issues.* Jones & Bartlett Learning.

Kosseff, J. (2018). Defining cybersecurity law. *Iowa Law Review, 103*(3), 985-1031. <https://search-proquest-com.ezproxy.liberty.edu/docview/2187899333?accountid=12085>

Lehman, C., DuFrene, D., & Walker, R. (2017). *ACP IAC MTR BCOM* (9th ed.). Boston, MA: Cengage Learning.

Satzinger, J., Jackson, R., & Burd, S. (2016). *Systems Analysis and Design in a Changing World.* Cengage Learning.

Stewart, C., Nodoushani, O., & Stumpf, J. (2018). Cultivating employees using Maslow's Hierarchy of Needs. *Competition Forum, 16*(2), 67-75. <http://ezproxy.liberty.edu/login?qurl=https%3A%2F%2Fwww.proquest.com%2Fscholarly-journals%2Fcultivating-employees-using-maslows-hierarchy%2Fdocview%2F2369806244%2Fse-2%3Faccountid%3D12085>

**Appendix A**

As stated throughout the paper, the PHP files strangely tend to return a blank page or random string of code despite having programming in place to prevent this issue.

Graphical user interface, text

Description automatically generated

Figure A1. The PHP page here is blank despite attempts to output error reports.

Graphical user interface, application, Word

Description automatically generated

Figure A2. For unknown reasons, this PHP page outputs portions of its code onto the screen.

**Appendix B**

Despite the complications shown above, the HTML pages remain as functional as they always have. They were only slightly modified in attempts to add compatibility with the new PHP files in this system.

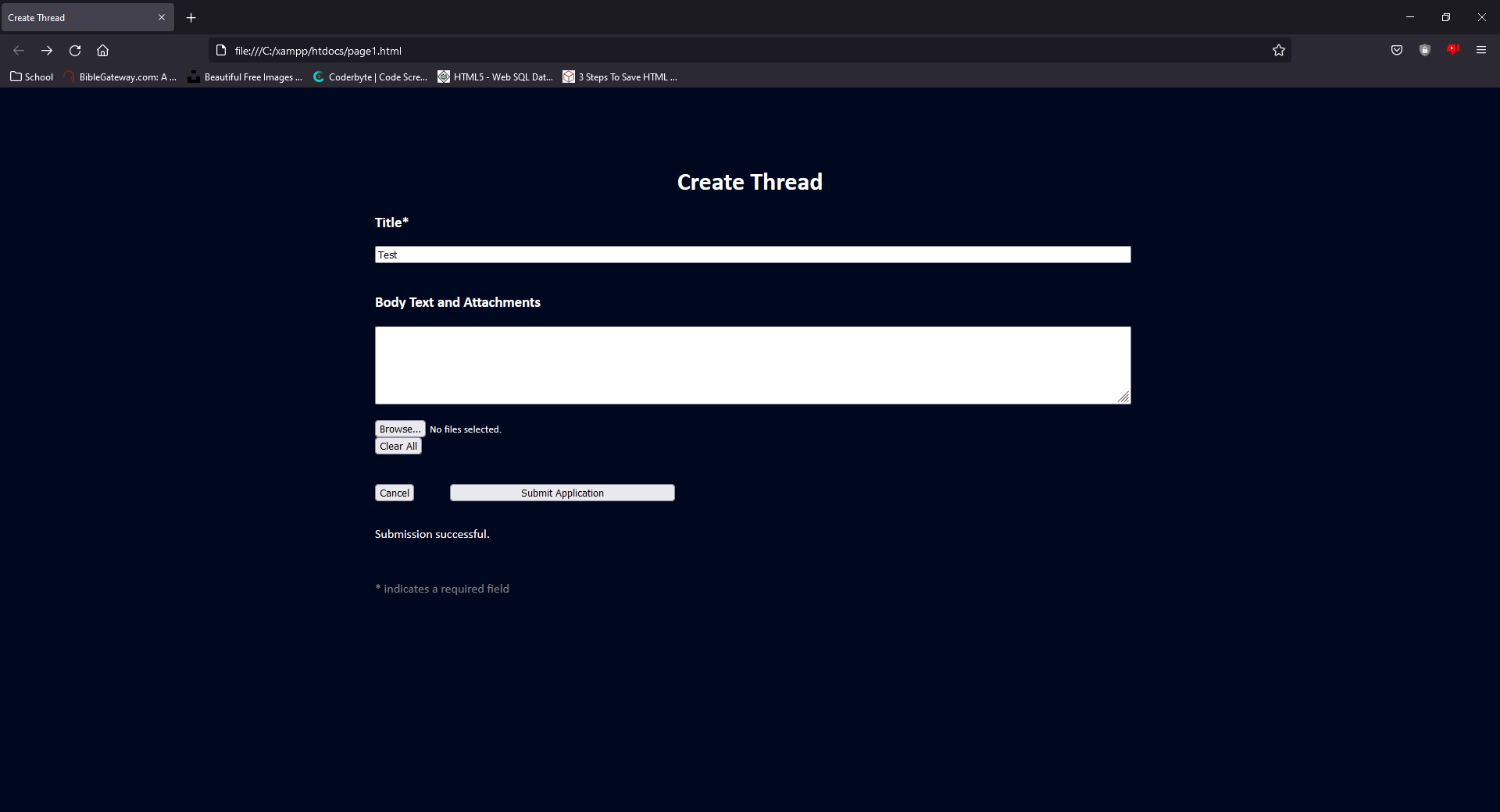


Figure B1. This screen allows users to attach regular threads onto the forum.

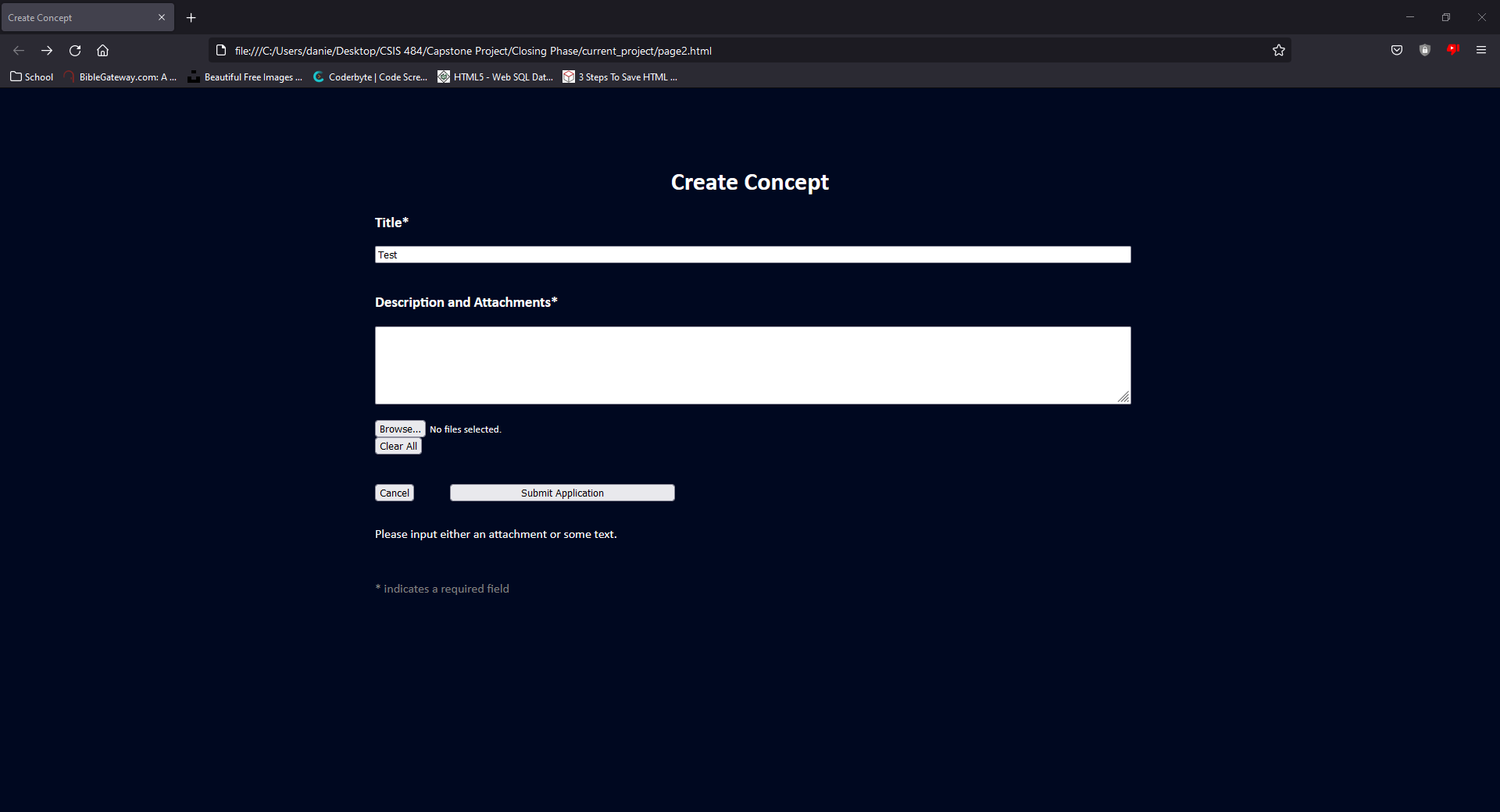


Figure B2. This screen appears almost identical, except it can upload concepts for new projects.

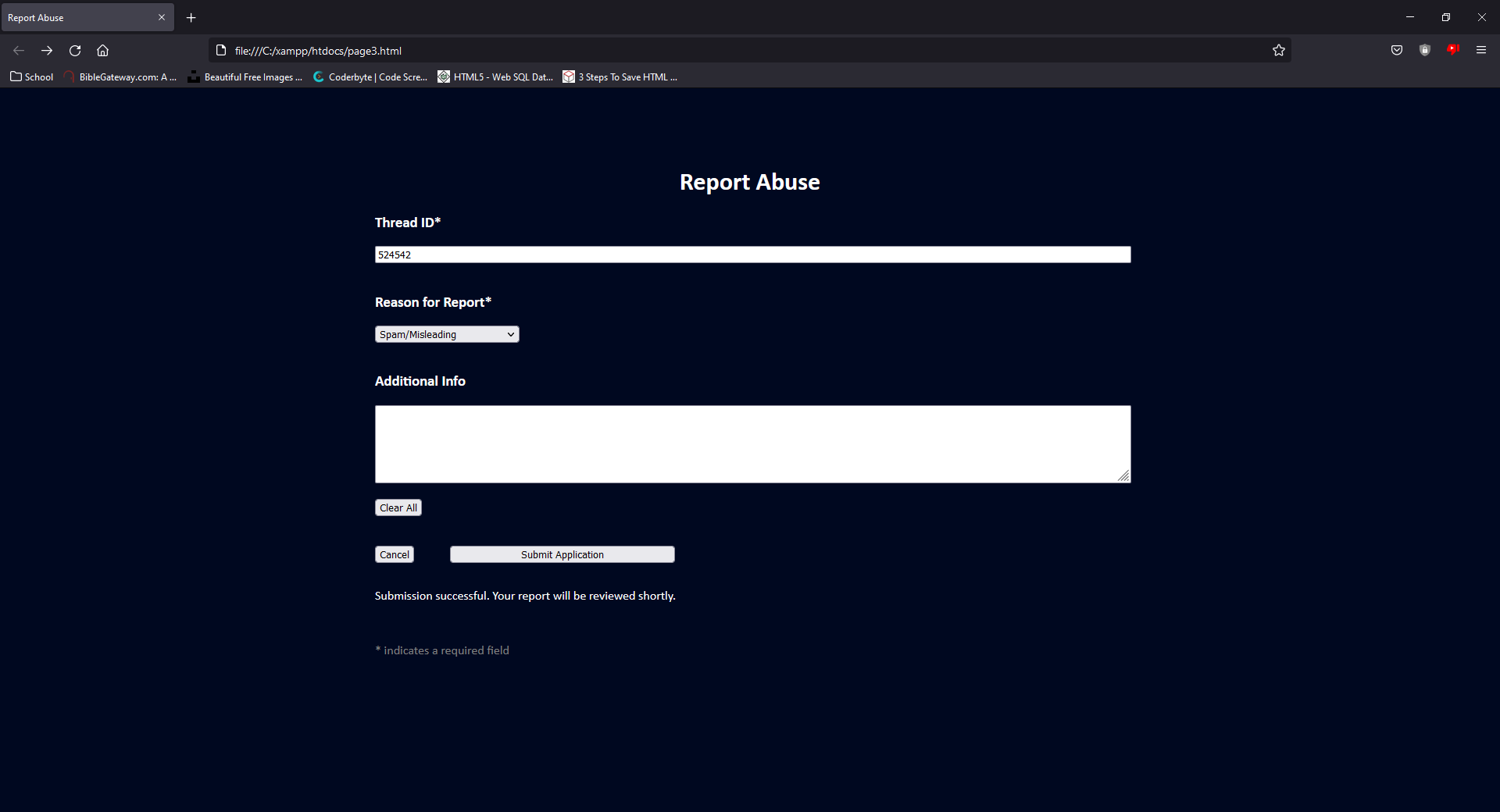


Figure B3. This page now allows users to specify which thread they wish to report, though the requirement to type an ID rather than the thread’s name would require the actual forum to display the thread’s ID before this system could be usable, let alone practical.

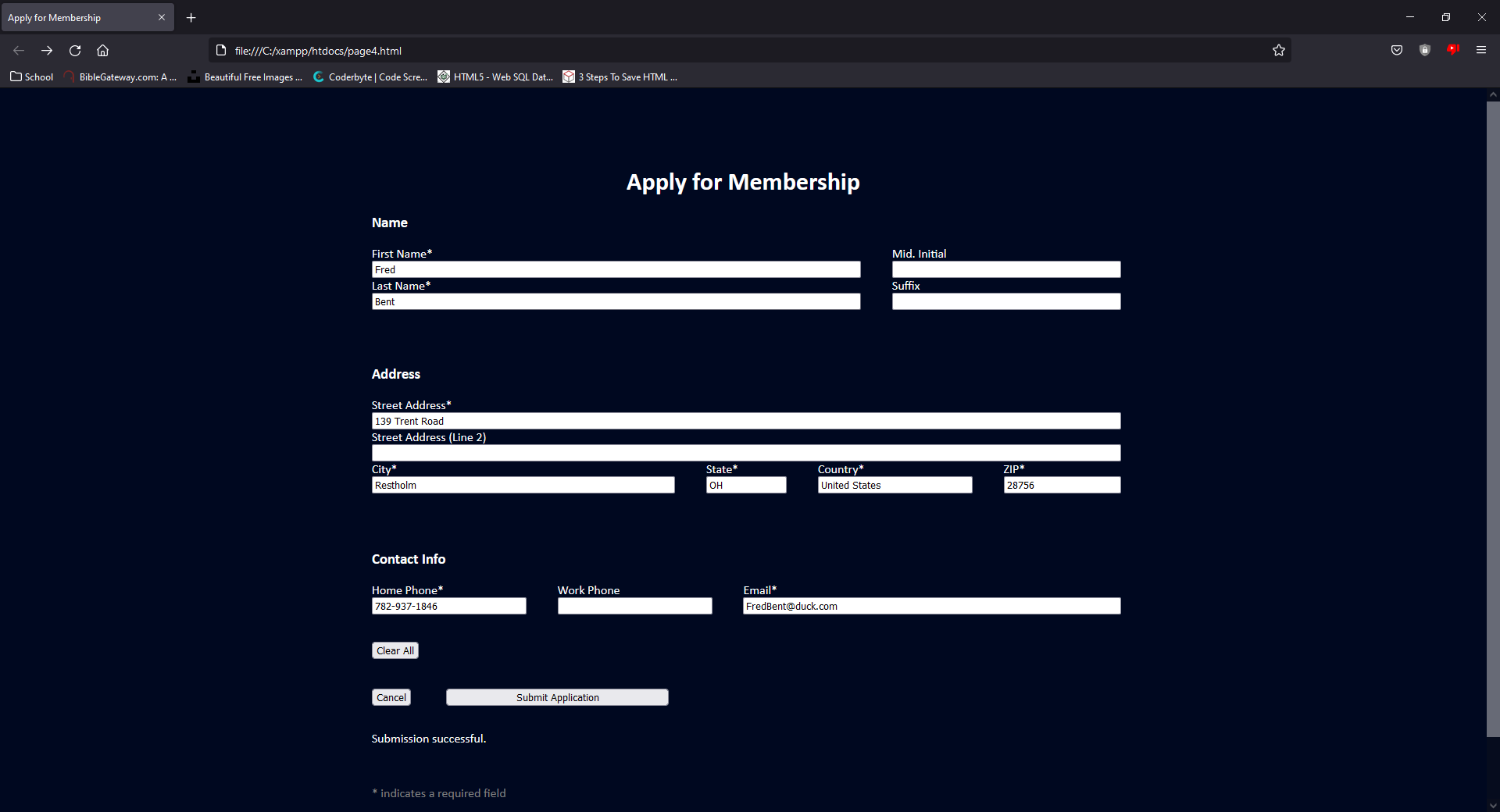


Figure B4. This page allows users to submit every credential necessary to verify their identity.