# CS 255 Module Two Assignment Template

## Functional Requirements

| **Functional Requirement** | **Rationale for Requirement** | **Source(s), APA format** |
| --- | --- | --- |
| **Login System with Real Security** | I don’t think you can run an LMS, or basically anything, in 2025 without multi-factor authentication. HTTPS, OAuth2, SSO. It needs to have safe logins, secure password protection, and vault-based storage (like using Argon2). That applies to admins, faculty, and students. | National Institute of Standards and Technology. (2022). *Digital identity guidelines (NIST SP 800-63-3)*. <https://pages.nist.gov/800-63-3/> |
| **GUI / Course Creation and Management** | Instructors need to have a class already made, or a way to construct their own classes through an easy interface/GUI, without calling IT. Drag-and-drop type features, modules, assignment uploads, grading policies, all in one self-service system. | D2L. (2024). Brightspace Learning  <https://www.d2l.com/brightspace/> |
| **Authentication & Authorization Layer** | **S**tudents should have no access to the teachers’ grade-books. Admins need ultimate access to control things like password resets, account permissions, etc. Password need to be strong, hashed, and securely stored. Teachers should be able to grade, email students, and manage assignments directly in the GUI. Clean, permission-based security system. | OWASP Foundation. (2023). *Authentication cheat sheet*. <https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html> |
| **Messaging + Discussion Zone** | Emails can be slow. Students and teachers need a built-in chat, announcement feeds like Brightspace, and forums. Forums were huge back in the day and still matter for discussion-heavy classes. Canvas gets this right. I’d probably combine the best parts from Brightspace, Canvas, and the rest. | Instructure. (2024). Canvas .  <https://www.instructure.com/canvas> |
| **Grade-book + Feedback System** | Professors grade. Students check scores. You need to have a working feedback system/loop or it falls apart. The LMS needs a grade-book with rubrics, support for bulk uploads PDFs, images, and other media and a comments system. Brightspace even has its own glitches so I’ve seen firsthand what happens when you don’t build this part right. | D2L. (2024). Brightspace Learning <https://www.d2l.com/brightspace/> |
| **File & Media Distribution Layer** | Content has to be interactive. All my apps are interactive, whether it’s in PyQt6 or JavaScript. Without interactivity, students won’t focus. The LMS needs to support HTML5, PDFs, embedded video, and ideally things like Jupyter notebooks or auto-grading code. If PyCharm can flag a mistake, so can an LMS. | **Jupyter Project. (2023).** Project Jupyter Documentation.  <https://jupyter.org/documentation> |

## Nonfunctional Requirements

| **Nonfunctional Requirement** | **Rationale for Requirement** | **Source(s), APA format** |
| --- | --- | --- |
| **Works on All Devices** | Every student uses different hardware, phones, tablets, laptops. If the LMS doesn’t display and respond cleanly across all those devices, the student experience will suck. This has to be a responsive UI, period. | **Instructure. (2024).** *Canvas LMS: Mobile Apps and Responsive Design.*  <https://www.instructure.com/canvas/features/mobile> |
| **ADA and WCAG Compliant** | There are some students that have different disabilities, and may need extra features when it come to coding the program so they can use it better. If the system isn’t accessible to everyone, we’re cutting out a entire part our student base. It needs to meet accessibility standards. | **World Wide Web Consortium (W3C). (2018).** *Web Content Accessibility Guidelines (WCAG) 2.1.*  <https://www.w3.org/TR/WCAG21/> |
| **Fast Response Time: < 2 seconds Load Time** | If pages are slow, students will not want to learn or be as motivated. Every action that the user does if its loading a quiz, doing an assignment, or sending an email, it should feel instant. Speed is usability | **Nielsen, J. (2010).** *Usability Engineering.* Morgan Kaufmann. |
| **Recovery and Backups** | One mistake shouldn’t delete any information for the student, or teachers or anyone using the program. We need daily backups, a type offsite storage, and rollback support for if something goes wrong. | **National Institute of Standards and Technology (NIST). (2020).** *Contingency Planning Guide for Federal Information Systems (SP 800-34 Rev. 1).*  <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-34r1.pdf> |
| **Scales** | Could be 1 or 10,000 students using it at once, but the backend has to scale. That’s the most important part of the full stack. If the backend can’t handle the load, the whole system fails. | **Amazon Web Services. (2023).** *Scalability and Elasticity in Cloud Computing.*  <https://aws.amazon.com/architecture/scalability/> |
| **Uptime Target = 99.9%** | If the LMS goes down mid-exam, anytime, it impacts every student’s ability to learn, submit work, or take a test, etc. This system needs redundancy, backups, and a fallback domain. Reliability isn’t optional to me for this when it comes to an application for a university. | **Amazon Web Services (AWS). (2023).** *AWS Well-Architected Framework: Reliability Pillar.*  <https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar> |

## Assumptions

| **Assumption** | **Rationale for Requirement** | **Source(s), APA format** |
| --- | --- | --- |
| **That students and instructors have access to the internet.** | Since the LMS is in the cloud we’re assuming users aren’t trying to submit homework on dial-up internet and have to have some type of stable internet connection. A system where you would need to complete this type of school work require a solid internet connection least for version 1 till it can be optimized in the future. | Pew Research Center. (2021). *Internet/broadband fact sheet*. <https://www.pewresearch.org/internet/fact-sheet/internet-broadband/> |
| **Faculty/Students will actually use the system** | I’m assuming instructors will actually build their course pages, update grades, and communicate through the platform, and that students will actually use the platforms such as the discussions posts that bright space has. | EDUCAUSE. (2020). *Faculty and student technology use in higher education*. <https://library.educause.edu/resources/2020/10/student-and-faculty-use-of-technology> |

## Limitations

| **Limitation** | **Rationale for Requirement** | **Source(s), APA format** |
| --- | --- | --- |
| **No offline mode** | LMS is designed to be an always-connected system. I don’t think a school could function any other way. Caching full courses, assignments, and syncing across devices adds a ton of complexity and security risk. Until there’s an actual reason to support offline mode. and a clean way to build it, this system will require an internet connection. | Siemens, G., & Long, P. (2011). *Penetrating the fog: Analytics in learning and education*. EDUCAUSE Review, 46(5), 30–40. <https://er.educause.edu/articles/2011/9/penetrating-the-fog-analytics-in-learning-and-education> |
| **No advanced intrusion detection or threat monitoring** | The LMS will have secure logins, encrypted data, and role-based access but it won’t ship with enterprise-level intrusion detection, or zero-day protection at launch. That stuff takes a lot of infrastructure. For now, I would just go with the solid best practices, but deeper threat detection would be something we’ll layer in later if the system scales. | Scarfone, K., & Mell, P. (2007). *Guide to Intrusion Detection and Prevention Systems (IDPS)* (Special Publication 800-94). National Institute of Standards and Technology (NIST). <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf> |