CAMPUSLINK

Team 38 - Product Backlog

Hareum Lee, Xavier Dextra, Sparkle Biswas, Divay Gupta, Joseph Jackson, Rudis Sprenne

Problem Statement

We are building CampusLink, a learning management system for creating, hosting, and editing online learning resources. Existing systems often have non-intuitive interfaces that can make it inconvenient for professors to perform routine tasks such as entering student grades, for students creating grade predictions, and for students to find assignments. For example, in BrightSpace finding quizzes and even assignments as a student is not always apparent and can be a hassle. We aim to make this process smoother with an easy-to-use interface.

Background Info:

Problem and Audience:

Today learning takes place beyond a physical classroom setting, and occurs wherever students have access to the internet. Students and instructors all over the world interact daily with their courses through the internet. Online learning platforms are widely used by students and instructors to teach and learn from anywhere, however these platforms can be unintuitive and difficult to use, and often multiple platforms are used to do the same thing.

Similar Platforms:

Several learning platforms currently exist, such as Brightspace and Canvas, and instructors often use supplementary services, such as Gradescope, Piazza, and HotSeat. Brightspace and Canvas are more complete learning platforms that allow students to turn in assignments, take quizzes, view grades, create discussions, and view course material. Gradescope is a service that allows students to turn in handwritten or typed assignments, either as pictures or pdf files. Piazza is a forum platform where students can ask instructors questions or participate in discussions, and HotSeat is a platform that also allows for discussions, as well as providing live polling.

Limitations:
One of the biggest limitations with the current complete learning platforms, such as Brightspace, is an unintuitive user interface. Students and instructors find it difficult to access assignments and quizzes, and depending on how the instructor sets up the course page the students might need to access assignments in different ways. We can fix this by creating a standard way to assign homework and other assignments, and ensuring all assignments can be found in one tab. Another limitation with the current system of learning is that multiple platforms are used, but the functionality of some platforms is already available in other services. We can address this problem by combining the functionality of multiple platforms into one platform and ensuring this one platform is intuitive to use and polished, so that instructors do not feel the need to use any other supplementary services.

Functional Requirements:

1. As a user, I want to be brought to the login page upon visiting the website.
2. As a user, I want to register for a CampusLink account.
3. As a user, I want to request for an instructor account on the login page.
4. As a user, I want to log in to my account on CampusLink.
5. As a user, I want to log out from my CampusLink account.
6. As a user, I want to be able to reset my password if I forget it.
7. I want to be able to access the website from multiple devices and have the pages scaled properly for each device.
8. As a user, I want to be able to navigate the website using a navigation bar.
9. As a user, I want to be able to view my account settings page.
10. As a user, I would like to easily navigate to features common to all courses, such as the calendar.
11. As a user, I want to be able to access and search a frequently asked questions page
12. As a user, I want to be able to edit my information on the account settings page.
13. As a user, I want to be able to post on the discussion board.
14. As a user, I want to be able to respond to a discussion post.
15. As a user (student), I want to receive update emails on any discussion posts I am a part of or have interacted with.
16. I want to have exclusive access to an admin page that displays all registered accounts and requests for instructor accounts.
17. As an admin, I want to be able to approve or deny professor account requests.
18. As a user (student), I want to be able to view past or submitted quizzes after the due date.
19. As a user (student), I want to be able to view my grades.
20. As a user (student), I want to be able to calculate my “what if” grade for all courses.
21. As a user (student), I want to be able to view my courses.
22. As a user (student), I want to be able to view course content.
23. As a user (student), I want to be able to take quizzes.
24. As a user (student), I want to be able to submit files.

25. As a user (student), I want to be able to enter the weights (temporarily) for the different graded items for each course.
26. As a user (student), I want to be able to login and have my courses be in student view.
27. As a user (student), I want to be able to see the names and emails of other students in the course.
28. As a user (student), I want to be able to bookmark assignments and lecture notes in a special bookmark tab.
29. As a user (student), I want to be able to view calendar page which shows due dates.
30. As a user (student), I want to get an email notification when the announcements page is updated.
31. As a user (teacher), I want to be able to create blank discussion boards.
32. As a user (teacher), I want to be able to create private or restricted discussions.
33. As a user (teacher), I want to be able to enter the assignment weights to the "what if grade" system.
34. As a user (teacher), I want to be able to create or delete courses I am teaching
35. As a user (teacher), I want to be able to modify the courses I created.
36. As a user (teacher), I want to be able to see a list of all courses I am teaching.
37. As a user (teacher), I want to be able to add students to the course sections I am teaching.
38. As a user (teacher), I want to be able to see the list of students registered for the course.
39. As a user (teacher), I want to create quizzes, such as multi-choice or short answer quizzes
40. As a user (teacher), I want to create assignment submissions.
41. As a user (teacher), I want to set due dates for assignment and quiz submissions.
42. As a user (teacher), I want to set the number of submissions for assignments.
43. As a user (teacher), I want to set the duration for quizzes.
44. As a user (teacher), I want to set the number of attempts for quizzes.
45. As a user (teacher), I want to see how many students submitted assignments.
46. As a user (teacher), I want to be able to post lecture slides or notes to the course.
47. As a user (teacher), I want to be able to upload videos to the course (such as lectures or other media).
48. As a user (teacher), I want to be able to embed videos to the course content (such as lectures or other media) using a URL.
49. As a user (teacher), I want to be able to login and have the page be in an instructor view.
50. As a user (teacher), I want to be able to enter grades for the students in each course.
51. As a user (teacher), I want to be able to compute and display a class average for each assignment and an overall grade.
52. As a user (teacher), I want to be able to update an announcements and calendar page.
53. (*If time allows*) As a user (teacher), I want to be able to view how many times a student has viewed the course page/content.

Non-Functional Requirements:

Architecture:
We will be building a web application that can be accessed on desktop and mobile web browsers. We will be using React JS to build our front-end and Firebase+Firestore for the backend (such as storing user information and credentials, course information, discussion board chats, etc.). We chose React for the frontend as it is a popular library with tons of support and functionality, which will aid us in better controlling the functionality we want to add to the web app. Firebase was chosen as it is a cloud platform, making it accessible from anywhere and also because Firebase offers real-time databases, authentication, and analytics on the performance of our web app (such as number of requests per second, time for each request, etc.). Firebase provides a NoSQL database called Firestore, which can be used to store and manage data in real-time. This data can be accessed by the frontend through the Firebase API.

Hosting and deployment:
Deployment will be handled using Vercel, which is a free-to-use platform that makes hosting easy and can deploy code directly from a GitHub repo. Overall, the architecture is designed to provide a scalable and flexible solution that can be easily modified and improved as the needs of the web app evolve over time.

Security:
To ensure the security of user passwords, we will be using a secure irreversible hashing algorithm such as bcrypt, SHA-256 or SHA-512 to store the passwords in the database. This will provide protection in the event of a security breach. To ensure that the passwords are strong and secure, we will be implementing strict password policies, requiring users to create passwords that meet specific criteria such as a minimum length of 8 characters, non-repeating numbers, not including a part of their email or name, and including at least one special character. Additionally, user sessions will be managed through the use of secure JWT tokens, and access to sensitive information will be controlled with access permissions, only allowing authorized users to access and modify the information they are permitted to, for example, students will not be able to modify grades. By implementing these measures, we aim to provide a secure platform for professors and students to interact and store their data.

Performance:
The website should be up 24/7 for all users, unless it is under maintenance, and it should be able to host at most a hundred users under the free Spark Plan of Firebase. This is only a financial limitation, and with additional payments, it is scalable and will be able to host as many users as needed.

Usability:

We strive to create an extremely easy-to-use and intuitive platform so that any student or professor can have an easy time utilizing the virtual classroom. There will be a variety of features, but they should all be easily understood, and be able to be utilized through smart user interface implementation. Students should be able to easily find and access quizzes, assignments, and lecture content. Professors should be able to update things such as grades, lecture notes, and student enrollment without much hassle. The platform should serve as an additional tool to benefit the class, rather than being an obstacle that gets in the way of efficiency.