PurdueHub - Project Backlog - Team 31

Aditya Gupta, Leonardo Delgado, Hoang Nguyen, Qasim Ali

Problem Statement:

Arriving at Purdue and beginning college is a significant change for new students. We believe there should be a single website that provides many functions that new students would need. Our website will combine features like the Scheduling Assistant, Boilerlink, campus maps and weather, making them available in one place for students to access easily. We will also allow students to create profiles, ask questions, and interact with each other.

Background Information:

Audience:

PurdueHub is created keeping Purdue University students in mind. As such, the main audience for PurdueHub are Purdue students, new and old. While PurdueHub aims to create an interactive, convenient interface to help new Purdue students manage and access their workload, we believe that the application's practicality and friendly interface make it a useful tool for older Purdue students to manage their schedule and workload.

Similar Platforms:

There exist several other student portal/scheduling applications that aim to help students manage their schedules and workloads. Some such applications include Shovel and Todoist. Shovel aims to help students manage their workloads by providing calendar and task-list features for organization. Todoist, similarly, provides calendar and to-do list features as well as different workspaces for each aspect of a student's life.

Limitations:

While both Shovel and Todoist, among the several other applications out there, provide calendar and task-list features, neither of them are specifically made for Purdue students. As such, these applications lack the useful functionality that Purdue APIs add

such as the Scheduling Assistant and BoilerLink. PurdueHub aims to solve this lack of functionality. By keeping Purdue students in mind, we can build an application with more features made specifically for Purdue students with official Purdue APIs. Additionally, PurdueHub will provide a social network of Purdue students making it easier to communicate with others on campus. Furthermore, the added functionality of the Scheduling Assistant, searching for events on campus, and browsing classes and clubs on campus give Purdue students more control over their campus life with PurdueHub.

Requirements:

Functional Requirements:

1. As a student, I want to be introduced to a basic tutorial of the general layout of the website, introducing tabs/information that are important to understand.
2. As a student, I want to be able to receive little tidbits of advice in the top corner of the application for important features that many other students miss out on - can silence these little notifications.
3. As a student, I want to be able to sign up for PurdueHub - including authentication, specific characters that are needed for my password, secure/encrypted storage of data.
4. As a student, I want to have my Purdue email account verified upon signup.
5. As a student, I want to be able to sign into PurdueHub after I have signed up.
6. As a student, I would like to be able to reset my password.
7. As a student, I want to be able to delete my account from PurdueHub with action confirmation.
8. As a student, I would like to be able to view my personal information on a separate page, such as username, bio, profile picture, etc.
9. As a student, I would like to be able to edit and customize my personal information including my profile picture, bio, username.
10. As a student, I want to be able to network with other students by being able to follow and unfollow fellow students.

11. As a student, I want to be able to block and unblock other students.
12. As a student, I want to be able to view a list of people I am following, a list of people I am followed by, and a list of people I've blocked.
13. As a student, I want to be able to direct text message fellow students for networking - chatting system.
14. As a student, I want to be able to delete chats or messages from the direct messages chatting system (if time allows).
15. As a student, I want to be able to specify who I want to be able to receive direct messages from: everyone or people I am following.
16. As a student, I want to report other students for inappropriate behavior.
17. As a student, if my account gets banned or deleted, I want to get an email notification or a notification when I try to sign in.
18. As a student, I want to be able to see mutual friends and club interests with other users.
19. As a student, I want to be able to be notified for certain aspects of information - professional development, club callouts, etc - implement a notification system, allowing users to turn on and off notifications.
20. As a student, I want to be able to view upcoming events on my homepage that I am interested in.
21. As a student, I want to be able to toggle events and clubs that I am interested in and want notifications for.
22. As a student, I want to be able to provide feedback on my experience on the PurdueHub experience.
23. As a student, I want to search and browse a list of available classes for the upcoming semester, including course details and schedules.
24. As a student, I want to easily share information about upcoming events, classes, and news with my friends and peers on social media.
25. As a student, I want a user-friendly search functionality that displays closely related classes from my search result (if time allows).
26. As a student, I want a user-friendly search that displays closely related usernames to my search request field (if time allows).

27. As a student, I want to be able to favorite classes and clubs to a personalized list for quick access.
28. As a student, I want to access a calendar feature - basic layout at first, empty, additional features added on top through other user stories
29. As a student, I want to be able to display important academic dates from the Purdue Academic calendar of the respective year applicable
30. As a student, I want to add final exams of classes that I am interested in or taking into my calendar view.
31. As a student, I want to connect my calendar and upload it from other calendar services such Google Calendar and Outlook calendar.
32. As a student, I want to download my calendar in a common format, such as .ical (if time allows).
33. As a student, I want to see an overall rating for each class and instructor based on student reviews. - pull from ratemyprofessor,reddit, etc
34. As a student, I want to view detailed class information, including course descriptions, prerequisites.
35. As a student, I want to filter and sort the list of classes based on my major, requirements, and class times to help me plan my course schedule.
36. As a student, I want to ask a chatbot general questions about Purdue (if time allows).
37. As a student, I want to view a map of Purdue using Google Maps and Purdue's maps.
38. As a student, I want to recommend and view recommended walking times between different buildings along campus.
39. As a student, I would like to get and report general campus updates, including construction, etc. (if time allows).
40. As a student, I would like to view the weather forecasted for that day in correspondence to my classes.
41. As a student, I want to view information about different buildings and draft information about such buildings (if time allows).
42. As a student, I would like to view an FAQ about certain topics related to purdue (if time allows).

Non-Functional Requirements:

Development:

PurdueHub will be developed using React for the frontend, which is a widely used, popular Javascript library, ensuring an interactive student experience with its robust features. For our backend, we will be using Node.js and Express.js to create our server-side infrastructure. We will manage our data that will be stored through the use of MySQL, a relational database system that excels at data integrity and performance. We will be accessing APIs from Purdue such as Scheduling Assistant, BoilerLink, Purdue Campus Maps, to integrate all these separate features into a cohesive product for students to easily access all at once. We are aiming to achieve a response time of less than a second for our authentication to get students quickly engaged in our application.

Security:

PurdueHub's storage of user credentials is a serious concern we will be addressing. By encrypting user information with a strong cryptographic hashing algorithm such as bcrypt, we will then store it in our database hashed. Bcrypt is a trusted library from Node.js, which when combined with the use of salting to add more complexity, will discourage attackers from brute-forcing their way into personal information from users. For our direct messaging feature, we will be using end to end encryption, along with the use of TLS to protect data from users and the server.

Response Time:

PurdueHub is aiming for the following response for respective sections from the application. If there is significant lag or delay in our services, it would impact the friendliness and usability the application aims to provide. Keeping this in mind, we aim for a latency of below 500 milliseconds for the messaging services provided. Additionally, any search functionality should be completed within 200 milliseconds. Changes in settings, preferred information or a friends list should take under 100 milliseconds. Finally, scheduling items in the calendar and syncing with other calendars, such as Google Calendar, should take under 1000 milliseconds.

Usability:

PurdueHub is aiming to create a user interface that is not only user friendly, but also easy to navigate especially when users try to access our application on their phone. We want to implement a smooth and responsive UI, that will be intuitive to easily access their saved information regarding classes, clubs, and upcoming news they want to be notified for. We want to design our interface to allow the user to access their profile/account page from anywhere in the web application within 4 clicks. We want to cater to visual impaired individuals, by including alt-text from non-text items on their page.

Hosting/Development:

The frontend and backend will be deployed separately as two different microservices. The frontend will be hosted for free on Github Pages, while the backend can be hosted, after we are far enough in development, on a cloud service provider such as AWS or Firebase.