Piper Academy

Introduction

We are a website-based teaching service that connects tutors and students for one-on-one tutoring sessions conducted via webcam and a shared IDE. We provide a secure, stable service with an intuitive interface and easy account creation and integration with Google. Students and tutors will be able to schedule sessions through an interactive calendar.

Requirements

Functional

- Design an online CS tutoring service with a video and text chat feature similar to other services such as Skype or Twitch.
- There will also be an IDE integrated into the tutoring session page that will allow all parties to see and write code simultaneously in Java or Swift. The IDE will also support plain text and mathematical symbols.
- Students will be able to book appointments with tutors via a calendar that lists tutors by subject. Will be 1-on-1 tutoring.
- The student will be able to toggle their camera feed on or off.
- The tutor will be able to switch between their camera feed showing their face or a blackboard (ie. switching between a front or back facing camera).
    - Both the student and the tutor will be able to compile the code and see the results.
- Setting up a login and registration page so students and tutors can set up different types of accounts tied to their preexisting Google accounts.
- Search function to find tutors by topic
- Come up with a way to store data on the backend (user info, scheduling data)
- A proper website with site information, navigation, a title page, feedback etc.

Requirements

Non-Functional

- Users' login information stays secure.
- Website should be visually appealing and easy to use.
- Interesting logo
- Rating System for teachers and students
- Hiring qualified teachers
- Proper Scheduling system between teacher and student

Sub-teams

Account Management

Front-End

Website template not responsive

- created new template

Making appropriate visuals for our product

- used “chalkboard/whiteboard” as a guide for design, made sure to keep everything clean and readable

Finding lightweight ways to implement features like responsive sidebars and integrating signup

- Only HTML and CSS are used, no Javascript and no code that could cause unnecessary lag

Video and Communications

Issues:

- Finding out what to do in order to provide camera, audio, and text support
- Integrating our code into the server

Solutions:

- Researching WebRTC and finding tutorials
- Working with the accounts management subteam
- Switching from node.js to flask

Back-End

Issues

- Communication issues between other teams
  - Teaching other team members to use our system
- Integration Process
  - Learn how to better cooperate with other teams to implement features

Solutions

- Before meetings, have a solid agenda and goals to accomplish by the end of the meeting
- Split tasks between the integration team members by knowledge base

IDE

- Issues
  - Communication between subteams
  - Getting compiling working on the website
    - Originally used firebase, ended up compiling directly on the server

- Solutions
  - Meeting with each sub team in order to communicate and integrate our work
  - For compiling, we worked with the back-end team to set up a compiler on the server

Lessons Learned

- Working as a team
- How important it is to communicate
- Time Management
- How to use GitLab
- Developed our programming skills
- The importance of using the Agile method