Detailed Design

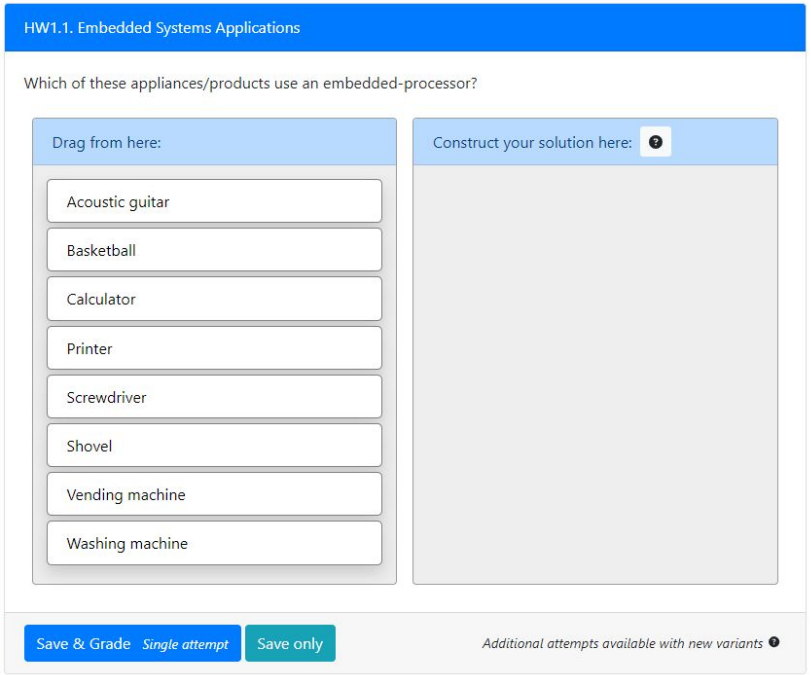
Team sdmay25-33

Advisor: Phillip Jones

Caden Otis, Devin Alamsya, Justin Cano, Joseph Krejchi, Rachel Druce-Hoffman

Project Overview

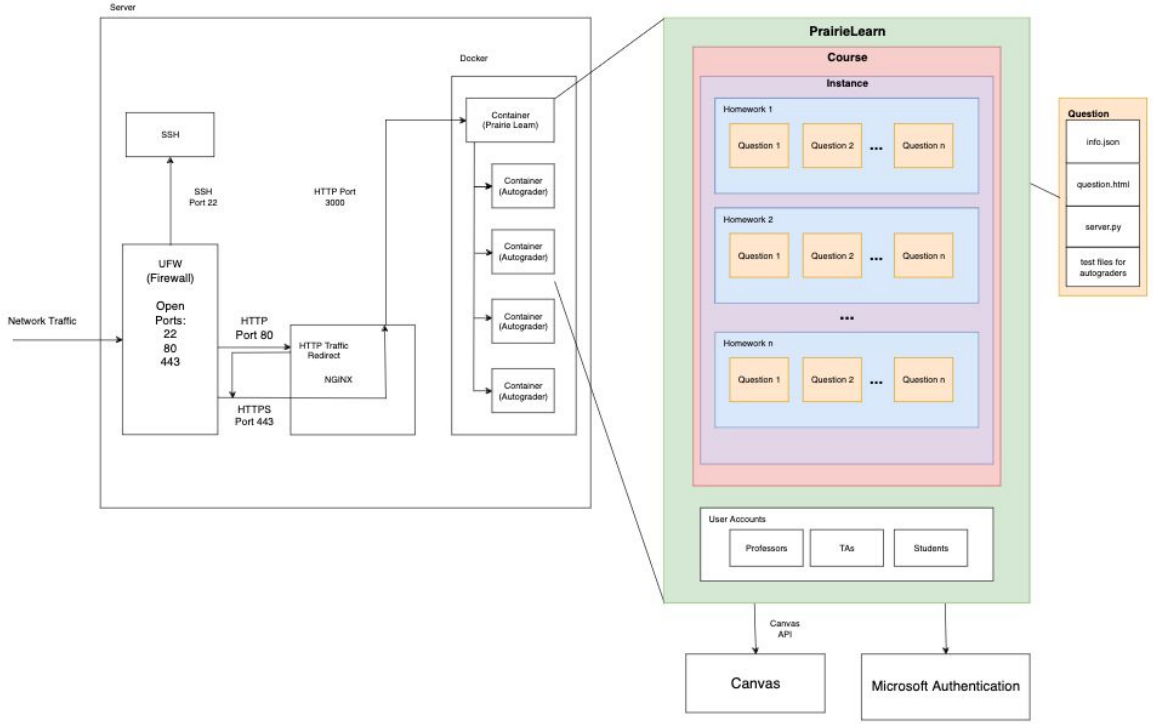
- Create an interactive application for CPRE 2880 students to better understand the concepts
  - HWs and quizzes
  - Randomized questions and autograding
  - Use emulation tools to simulate microcontrollers
  - Potentially have an emulated Cybot robot interface
- PrairieLearn framework to host the application
- Utilize Python, JavaScript, C and other programming languages
- Hope to inspire other professors to build similar interactive tools for their students



Problem Statement

- Students don’t get enough practice of concepts
    - Little feedback on Canvas HW submissions
- Not always availability to practice programming on the microcontroller in the lab
- Limited time to meet with Professor and TAs
    - Lab, class, office hours
- Limited capabilities with Canvas platform

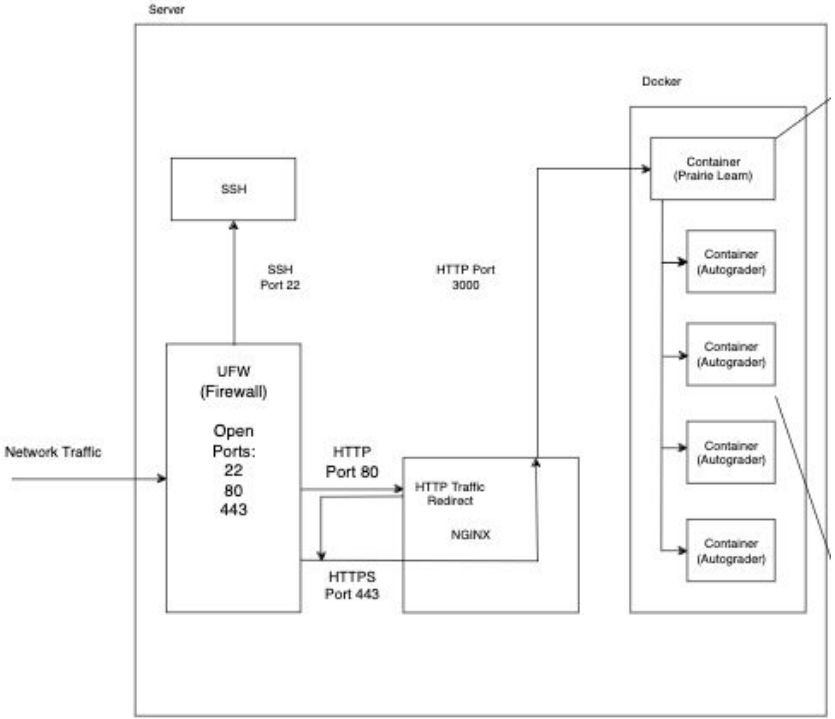
Detailed Design and Visuals Overview



Detailed Design and Visuals (1)

Server:

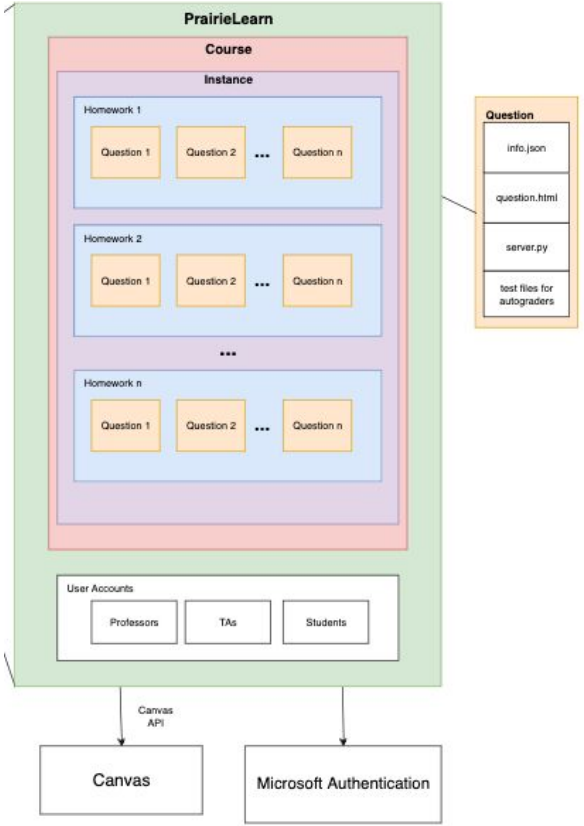
- Contains several security measures
  - Firewall, NGINX, and SSH security
- Connections allowed through ports 22 (SSH), 80 (HTTP), and 443 (HTTPS) only
- Network traffic gets encrypted when going to PrairieLearn container
- Houses several autograder containers for simultaneous question grading



Detailed Design and Visuals (2)

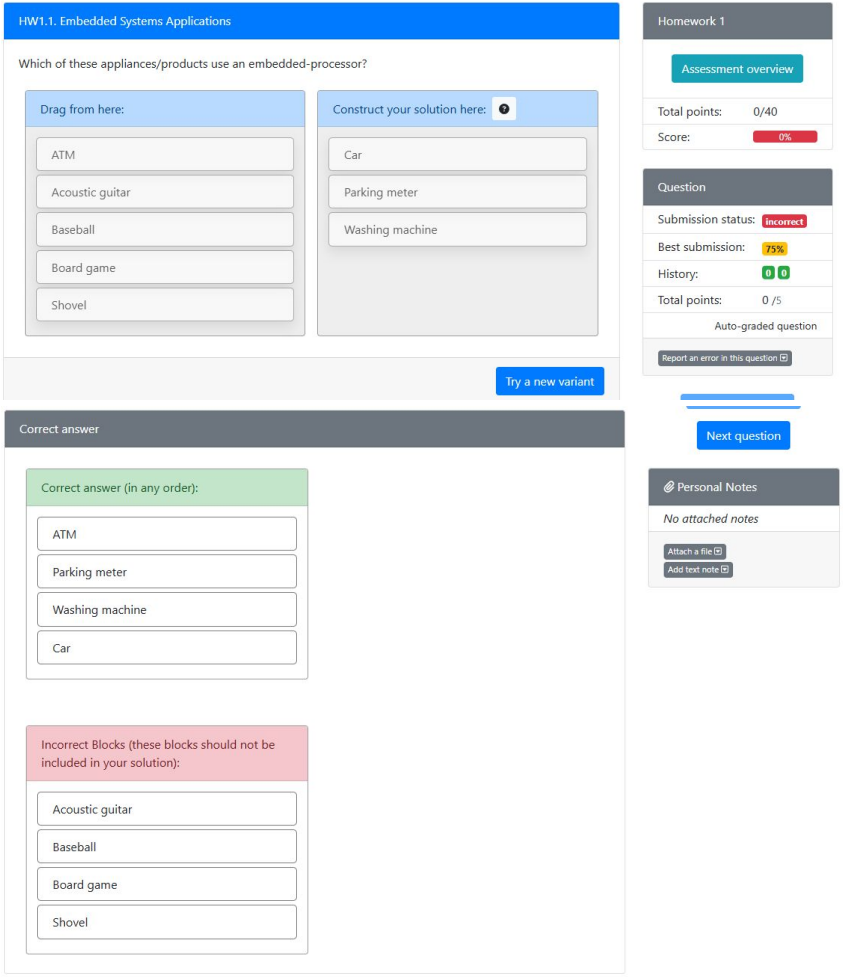
PrairieLearn:

- Allows users to create online courses with question sets, automatic grading and feedback, and gradebook integration
- Authentication for ISU students and professors
- Instructors can create courses with specific instances for each semester
- Students are provided with an interface that makes it easy to learn and engage with the questions
- Info.json handles question configuration
- Server.py handles question generation and randomization
- Question.html handles the UI of questions



Functionality

- Course instance will be reachable from Canvas
- Our project will be interactive and user friendly
- Questions will be randomized for every student
    - Students can get additional randomized instances of questions for more practice
- Students will answer questions in a variety of formats
    - Multiple choice, matching, writing code, etc.
- Most questions will be autograded for instant feedback
    - Professors and TAs can give additional feedback as well
- Grades will auto populate in the Canvas gradebook



Technology Considerations

- Virtual Machines
  - ETG-issued
  - Personal VMs for development
  - Ubuntu 22.04
- Git, Gitlab
  - Version control
  - Repo attaches to server

Areas of Concern and Development

- Legacy project: in development but doesn't fit user needs yet
- Areas of concern
  - Visual cohesiveness
  - Adequate questions
    - Complex, engaging coded elements
  - ISU authentication integration
  - Security

Conclusion

- Detailed understanding of project is crucial to implementation
- Our project is almost all software-based
  - Many individual code pieces to have work together
  - Different coding languages
- Through knowing our design’s intricacies, we can plan development better and prevent errors

Any Questions or Suggestions?