Group 40

Final Presentation

Introduction

Our **web based application** is an attempt to **bridge the gap** in communication by designing a convenient way to interact in a **live lecture** amongst a **large classroom**

Market Research

Piazza:

- Expensive
- Missing live features

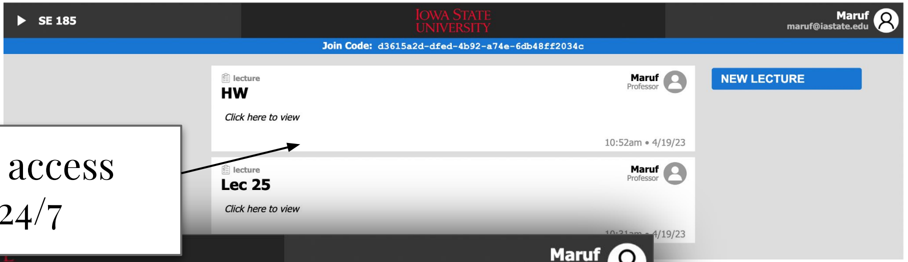
Tophat:

- No message feature
- Only professor to student interaction

Discord:

- Distracting
- Poll isn’t seamless
- Not education focused
- Channel categorization not built for lectures

Our Project in Action

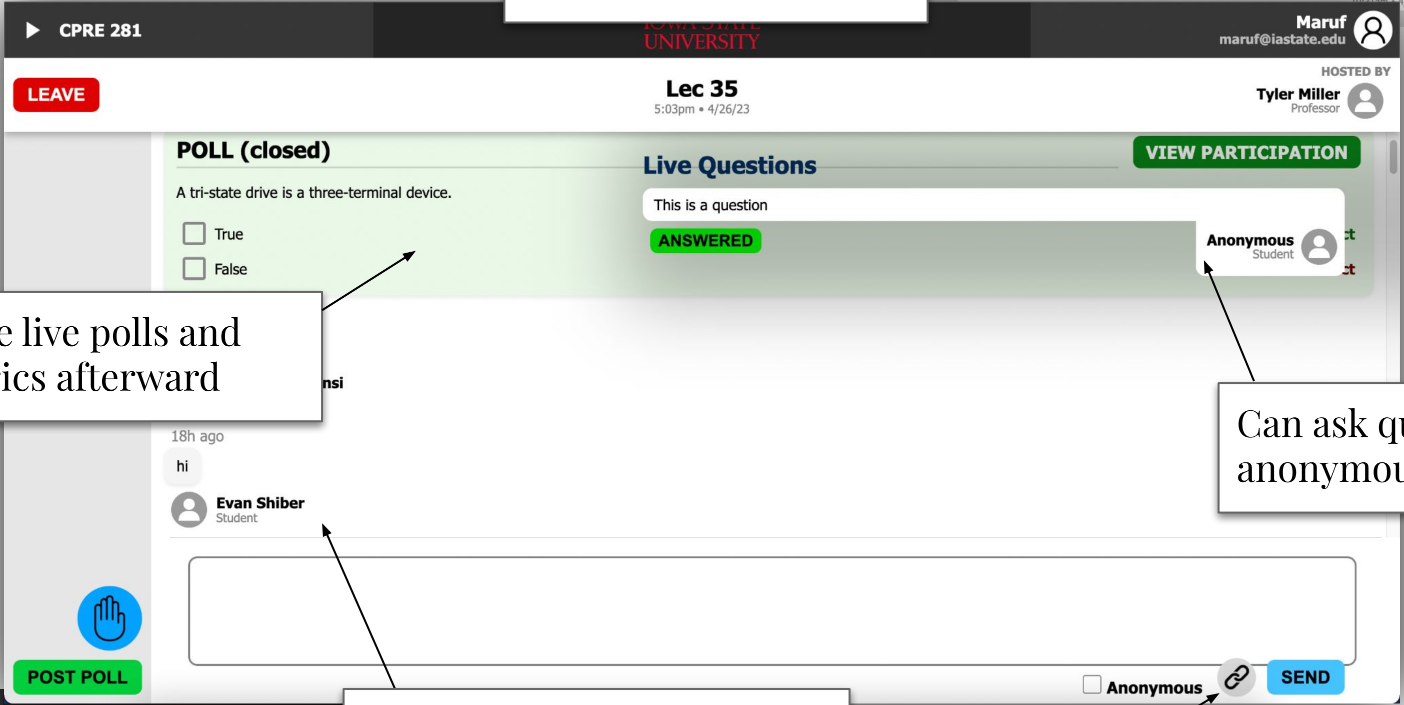


Join Code: d3615a2d-dfed-4b92-a74e-6db48ff2034c

lecture
HW
Maruf Professor
Click here to view
10:52am • 4/19/23

lecture
Lec 25
Maruf Professor
Click here to view
10:31am • 4/19/23

Can view and access past lectures 24/7



Lec 35
5:03pm • 4/26/23

Tyler Miller Professor

POLL (closed)

A tri-state drive is a three-terminal device.

- [ ] True
- [ ] False

Live Questions

This is a question
ANSWERED
Anonymous Student

18h ago
hi
Evan Shiber Student

Can create live polls and view metrics afterward

Can ask questions live anonymously

Can chat with the professor and other students

…including attachments

Implementation - Frontend

- React
- React Router
- Babel
  - For processing JSX
  - Improving capability
- Webpack
- Cypress
  - For testing

- Important utilities:
  - Axios
  - Redux

Implementation - Backend

- Node
- Ubuntu VM
- Express.js
    - Rest APIs, and Websockets
- SSL/TLS
    - (https, wss)
- MySQL via Docker
- Pipeline for deployments via Gitlab Runners
- Server running at sdmay23-40.ece.iastate.edu

Work Accomplishments - Frontend

- Pages (100%):
  - Signup (100%)
  - Login (100%)
  - Home Page (100%)
  - Live Lecture (100%)
  - Cataloged Lecture Page (100%)
  - Profile Page (100%)
  - Poll Submission Pop-up (100%)
  - Course Addition Pop-up (100%)
- Testing (90%)
  - UI testing (100%)
  - integration testing (90%)
- Remaining work
  - Add more testing

Work Accomplishments - Backend

- APIs:
  - Data insertion (100%)
  - Data fetch (100%)
  - Data edit (85%)
  - Data delete (85%)
- Pipeline (100%)
- Websockets (100%)
- File upload/storage/download (100%)
- Database (100%)
  - Schema Design (100%)
  - Creation (100%)

Key Contributions - Frontend

Adam Walters:

- Frontend code + build structure
- React topic expert
- Majority of frontend API integrations
- Live-lecture streaming
- Interactive messages + polls
- Course view page
- Lecture creation
- Generic pop-up form component
- Comprehensive website styling overhaul prior to demos

Jaden Ciesielski:

- Add course pop-up + API
- Poll pop-up + API
- Generic pop-up extensions
- Spinner component
- Profile page + patch API for editing user fields
- Signup page + API
- Home page + automatic refresh when course is added

Brandon Burt:

- Add login page + API
- Poll participation view
- Student pill component to list student poll results
- Buttons/reusable components
- Hover UI for components
- Editing messages/polls with elevated user permissions
- Wrote frontend tests with Cypress

Key Contributions - Backend

Tyler Miller:

- Backend application security
- Majority of unit and integration testing
- Created Deployment Pipeline
- Created a majority of the APIs
- Created the Websocket handling
- Wrote most of the backend API docs (OpenAPI and Websocket docs)
- Created process for running the app locally
- Handled database creation and updates
- Handled ISU Server setup/config

Alex Swenson:

- Database schema
- Various creation and data editing APIs
- File upload/download
- File storage
- Backend linting
- Contributed to backend API documentation

Guan Lin:

- Several APIs
- Helped with database schema
- Contributed to API documentation
- Unit tests

(For Frontend)

- Frontend poll results .csv download
- Some website styling

Challenges and Solutions

Frontend

- Difficulty maintaining to uniform design
    - Assigned one person to cleanup
- Making the site intuitive
    - Demoed & used classes to test our site and see where improvements could be made
- Lack of manpower / topic experience
    - Pull people from backend when needed
    - Spent additional time in the design phase to learn React

Backend

- Setting up backend locally
- Setting up deployment Pipeline
- Integration testing
    - Jest initially, switched to Postman
- Security issues
    - No Cybersecurity major students

Challenges and Solutions

Team

- Connecting APIs
  - When backend creates an endpoint, connects to it ASAP through Postman to promptly work out issues
- Deciding priority
  - Create our priority and confirm/make updates depending on client's thoughts
  - Tried to use an Agile development approach

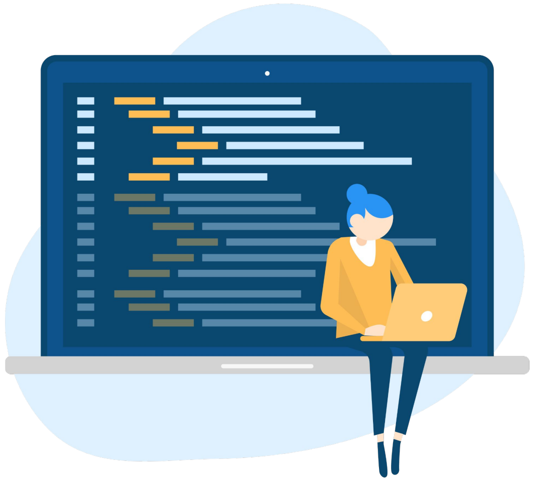
Future Work

Frontend:

- Add more tests:
  - Interface and end-to-end testing
- More post types beyond lectures
  - e.g. Announcements, Questions, Assignments

Backend:

- Better integration-style tests
- Supporting any additional features



Conclusion

- Very successful project and demonstrated practical use in real classes
- Completed just about 99% of everything we wanted to accomplish within the time frame that we had
- Hope for future work to be completed and later used in multiple courses