



SmartClass

...



Team #:
sddec25-10

...

The diagram shows three circular nodes connected by lines. The top-left node is a small blue circle with a white outline. The top-right node is a medium blue circle with a white outline. The bottom-right node is a large blue circle with a white outline. Lines connect the top-left node to the top-right node, the top-right node to the bottom-right node, and the top-right node to the bottom-right node. There are three dots above the top-left node, three dots above the top-right node, and three dots below the bottom-right node.

Team Members:
Josh Dwight, Ryan Lin, Logan
Pfantz, Matthew Gudenkauf,
Michael Geltz, Michael Becker

Client/Advisor:
Md Maruf Ahamed





01 SmartClass Mission



Problem Statement

The SmartClass project aims to address the challenge of student engagement in large classrooms. Instructors often struggle to facilitate active participation, as students (especially the ones seated farther back) find it difficult to ask questions or contribute to class discussions. Traditional methods like verbal Q&A sessions or raising hands can be inefficient in large settings.

What is SmartClass?



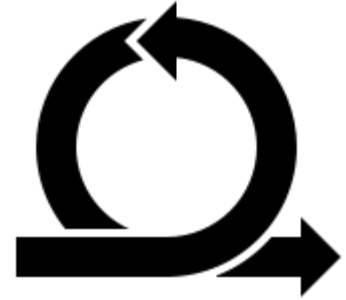
What are we creating?

- An interactive classroom platform that connects students and teachers
- Real-time polling and Q&A system for immediate feedback
- Interactive lectures activities to ensure student engagement
- Discussion space that persists beyond class time
- Analytics dashboard for tracking student engagement

Who are our intended users?

- Students
 - Shy - Doesn't like to ask questions
 - Confused - Professor went too fast through content
- Professors
 - Large classroom
 - Wants to make lectures engaging
 - Wants feedback

Development Style



● Agile Workflow

- Weekly faculty advisor meetings
- Weekly meeting with all team members
- Decided what needed to be implemented & what needed change



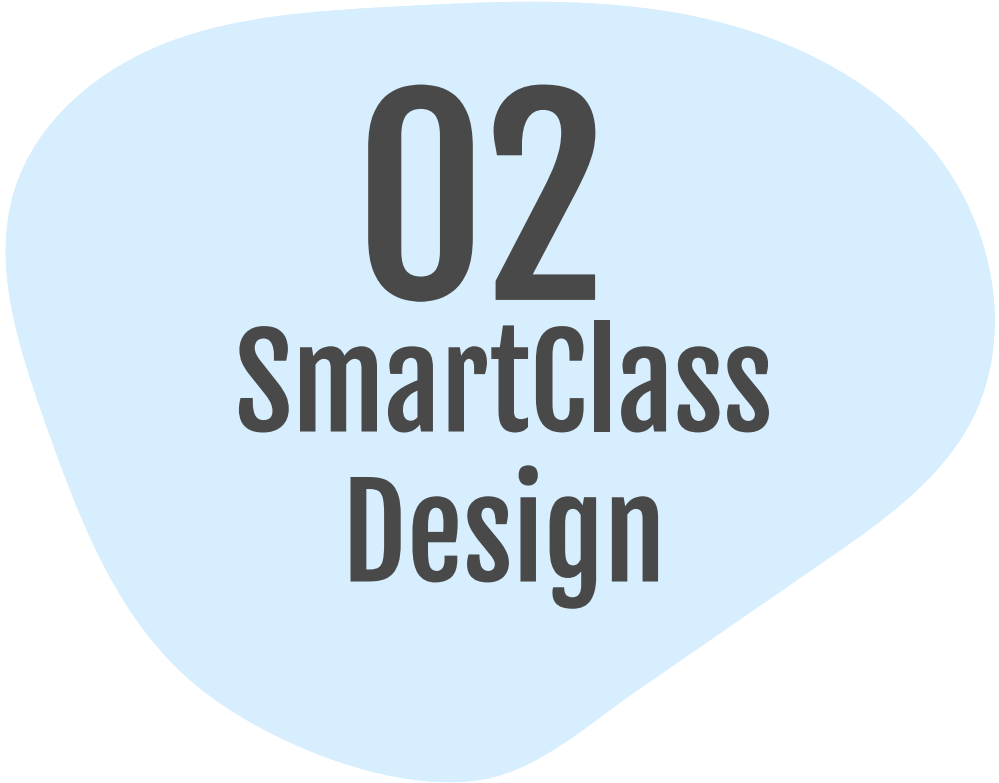

● Sprint Work Week

- 1 week long sprints
- Utilized GitLab issues for organization

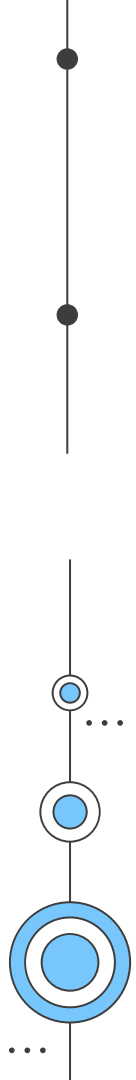


● Collaborative Environment

- All ideas valued and considered



02 SmartClass Design



Initial Mockups

SmartClass
Hello, Kelly ▾ Sign Out

Home Classes Chat Schedule People Support Search About

Upcoming Lectures:

COMS:352 | 02/14/25 9:30AM - 10:20AM:
 SE:421 | 02/14/25 10:30AM - 11:20AM:
 COMS:352 | 02/17/25 9:30AM - 10:20AM:

[\(View more...\)](#)

Previous Lectures:

SE:419 | 02/13/25 2:00PM - 4:00PM:

Lecture Notes:

Recap

Pages
Address space and physical memory are divided into equal-size units: pages and frames. A page table maps from VPNs to PFNs for each process.

- Translation:

Concern

Each process has its own page table which the OS stores in frames in physical memory.

- Linux page size is 4KB, so you'll see the address space is 1073B -- 17 million page table entries for every process!
- Linux reduces size with multi-level (3-level tree structure) page tables
- Page table lookups to show every memory access requires additional memory accesses! How is speed of memory?

L10: Translat

Jessica: What does VPN mean?
 Answer: Virtual Page Number

Henry: What does PFN mean?
 Answer: Physical Field Number

Quizzes:

- For each of the following memory allocation schemes explain why it does or does not experience external fragmentation or internal fragmentation.
 - a. Base and bounds
 - b. Segmentation
 - c. Paging
- Star system: (don't program to allocate more memory to its address space during execution. Allocation of data in the heap segments of programs is an example of such allocated memory. What is required to support dynamic memory allocation in the following schemes?
 - a. Base and bounds
 - b. Segmentation
 - c. Paging
- For each of the following virtual addresses, compute the page number and offset when using 4KB pages.

page 0
range 1

0 1

frame number
0
1
range 0

Score: (8 / 10)

SmartClass
Hello, Kelly ▾ Sign Out

Home Live + Classes Chat Schedule People Support Search About

COMS:352 | 02/14/25 9:30AM - 10:20AM:

Live View:

Recap

Paging

- Address space and Physical memory are divided into equal-size units: pages and frames
- A page table (mapping from VPNs to PFNs) for each process
- Translation:

Page Table ←

Virtual Address	0	1	C	1	0	1
VPN	0	1	C	1	0	1
offset						

Address Translation

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

Physical Address	1	1	1	C	1	0	1
PFN	1	1	1	C	1	0	1
offset							

Incoming Quiz...

Jessica: What does VPN mean?
 Mark (Prof): Virtual Page Number

Mark (Prof): What is the Big-O of this function?

SmartClass
Hello, Kelly ▾ Sign Out

Home Live + Classes Chat Schedule People Support Search About

COMS:352 | 02/14/25 9:30AM - 10:20AM:

Live View:

- For each of the following memory allocation schemes explain why it does or does not experience external fragmentation or internal fragmentation.
 - a. Base and bounds
 - b. Segmentation
 - c. Paging
- Most systems allow a program to allocate more memory to its address space during execution. Allocation of data in the heap segments of programs is an example of such allocated memory. What is required to support dynamic memory allocation in the following schemes?
 - a. Base and bounds
 - b. Segmentation
 - c. Paging
- For each of the following virtual addresses, compute the page number and offset when using 4KB pages.
 - a. 22,000
 - b. 77,056
 - c. 197,012

page 0
range 1

0 1

frame number
0
1
range 0

Changes in Development

SmartClass Student Dashboard lwpfantz ▾

Dashboard

[Join a Class](#)

Math 1302

Subject: Math


Description: Math 1302 provides a foundational exploration of algebraic

Example Class

Subject: Not set

Uploading materials 101

Subject: Not set



Math 1302

[Live Chat](#)

[Announcements](#) | [Discussion](#) | [Quizzes](#) | [Students](#) | [Materials](#) | [Settings](#)

Course Announcements

Exam 2 on Thursday (Nov 20)

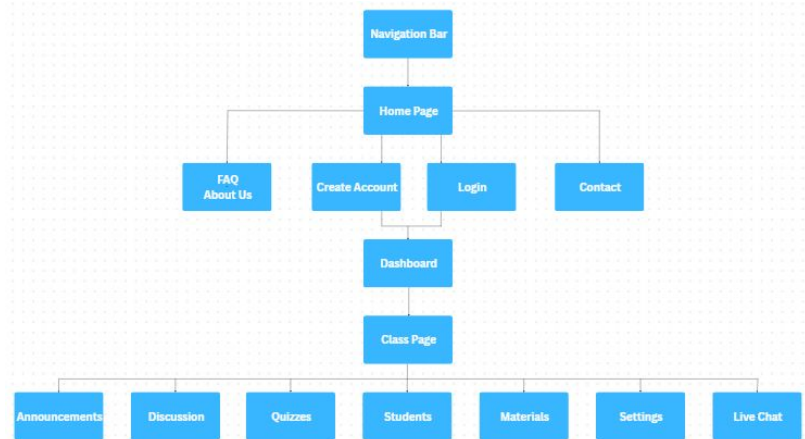
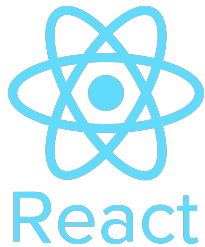
Hi all, just a reminder that the second exam is coming up on Thursday (Nov 20). Same format as Exam 1: in class, one 8.5x11 sheet of paper for notes, etc. Following the exam, we will of course have the Thanksgiving break and then we'll dedicate the last four regular course meetings to your projects.

Posted: 12/9/2025, 2:56:46 AM

Matthew Gudenkauf

Frontend

- React JS
 - Component-based architecture
 - Reduces redundancy through reusable UI elements
 - Enables dynamic, highly interactive pages
 - Supports real-time updates such as quiz notifications
 - Persistent navigation bar across all pages
 - Login required to access class content



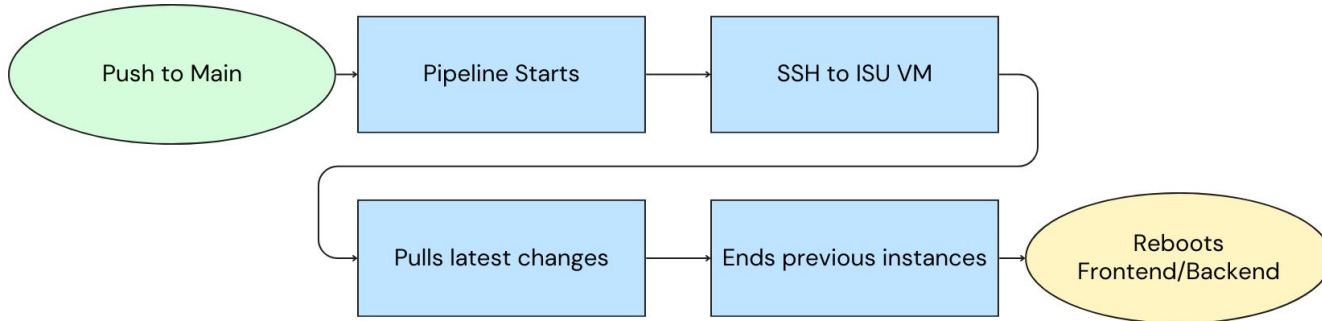
Backend

- Java & Spring Boot
 - Our backend code is written in Java.
 - Spring Boot for its web development features, security, authentication, and automated tests.
 - We utilize Maven for managing dependencies.
- MySQL
 - We use a MySQL server hosted on our own dedicated server.
 - MySQL allows us to appropriately design complex relationships and perform difficult queries.



CI/CD

- GitLab CI/CD
 - Autonomous deployment
 - Utilizes YAML file to run scripts
 - The deployment script follows these actions



Design

Course Creation
and Management

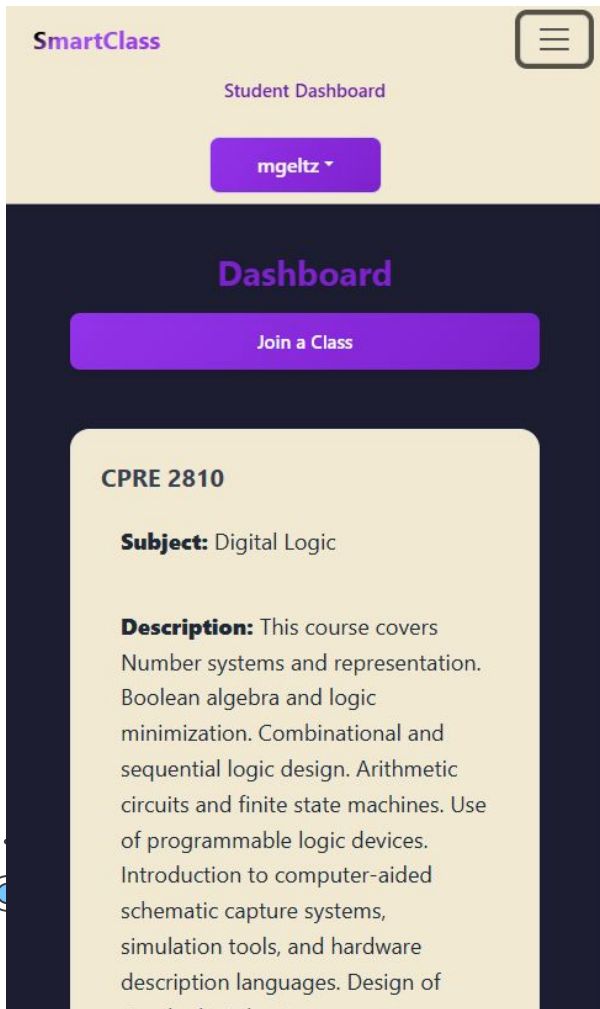
Live Polling and
Questions

Anonymous Question
Submission

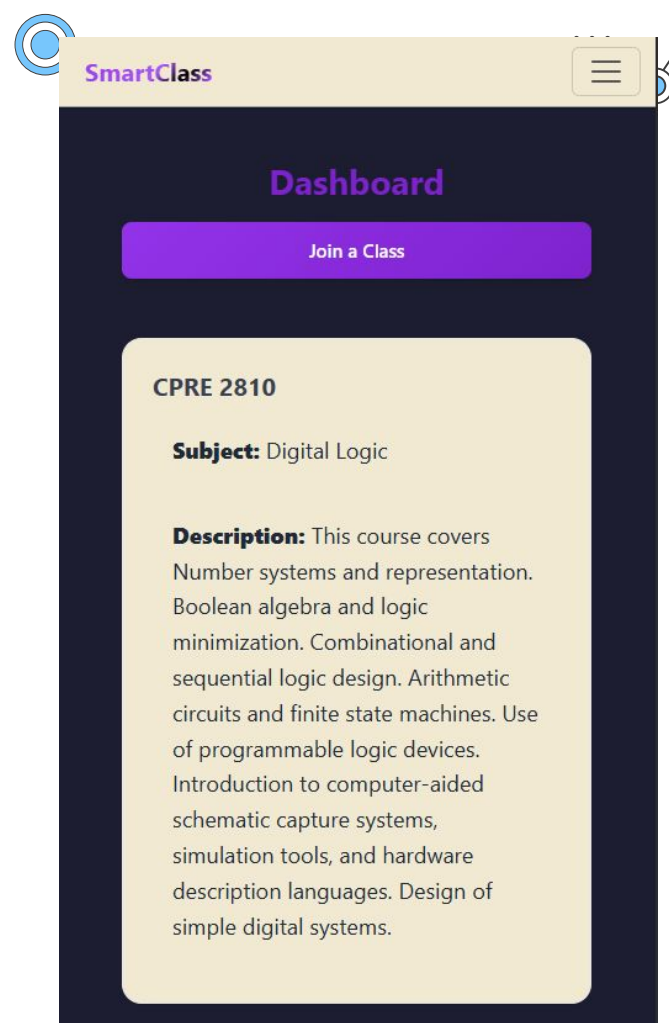
Class Material
Distribution

Student Participation
Tracking

In-Class Discussion
Forum

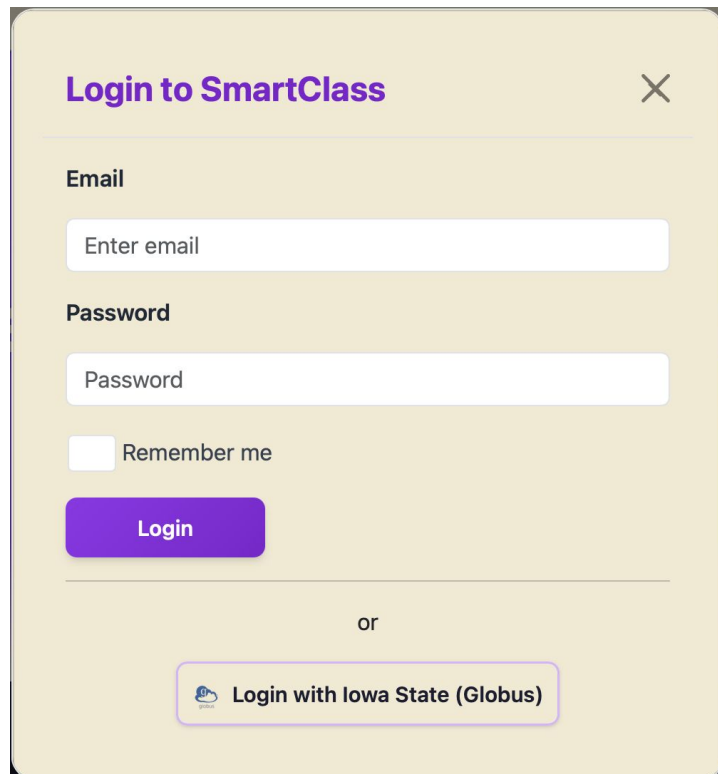


Compatibility



Management Features

- Create Account
- Log in
 - Account info page
- Create a Class
- Join a class



The screenshot shows a login modal window titled "Login to SmartClass" with a close button (X) in the top right corner. The form contains the following elements:

- Email:** A text input field with the placeholder text "Enter email".
- Password:** A text input field with the placeholder text "Password".
- Remember me:** A checkbox followed by the text "Remember me".
- Login:** A purple button with the text "Login".
- or:** A horizontal line with the word "or" centered below it.
- Login with Iowa State (Globus):** A button with the Globus logo and the text "Login with Iowa State (Globus)".

Classroom Features

- Announcements
- Quizzes
- Materials
- View their Students

Math 1302 Join Code: yhoS0z

Announcements

Discussion

Quizzes

Students

Materials

Settings

Features

- Discussion forums

Lecture 1 comments 12/9/2025, 3:24:01 PM

Asked by: Matthew Gudenkauf

[Reply](#) [Hide Thread](#) [Edit](#) [Delete](#) [Participation](#)

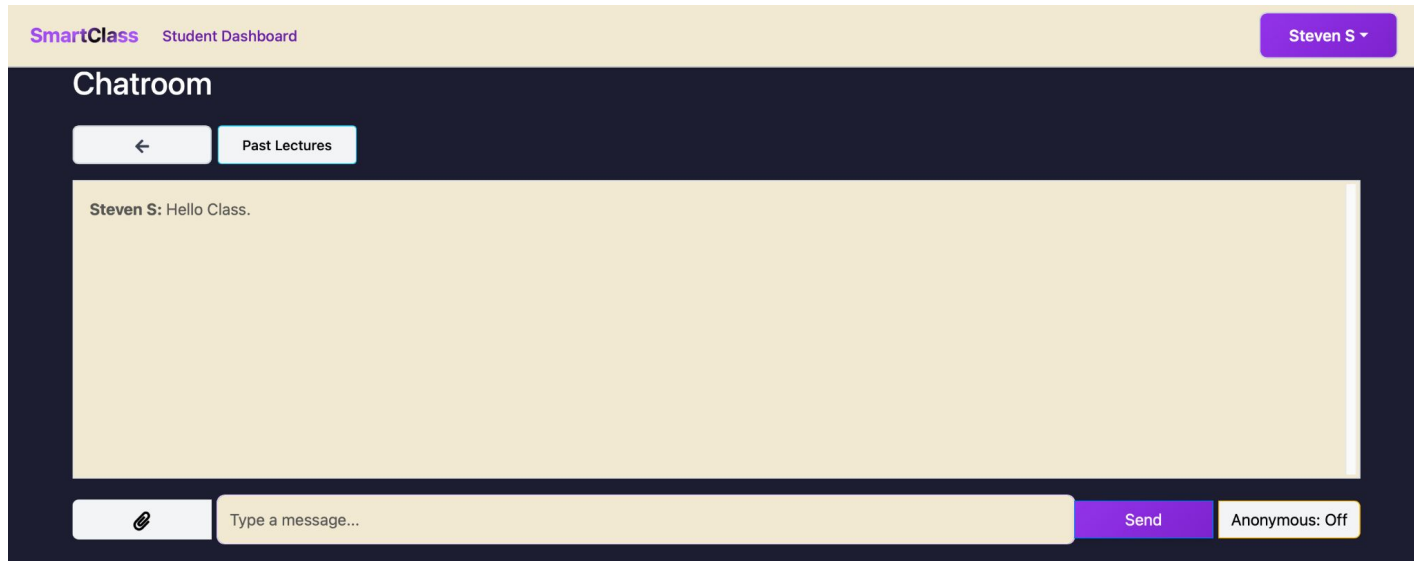
Can you explain? 12/9/2025, 3:25:04 PM

Replied by: Steven S.

[Reply](#) [Delete](#)

Features

- Live chat



Future Development

- AWS integration
- Support more file types
- Expand to multiple universities
- Commercial Product



Non-Functional Requirements

- Performance
 - UI should be responsive and consistent
- Reliability
 - Data should have backups
 - Servers should be capable of operating under expected load
- Scalability
 - Database should be uncomplicated, clean and consistent relationships
 - Code base should be well documented and easy to understand
- Compatibility
 - Ensure our web application functions properly on all OS and browsers.
 - Drop support for older & known-to-be vulnerable browsers to ensure security and modern feature functionality.



Testing/Security



Unit Testing

- Frontend
 - React Testing Library
 - Jest
- Backend
 - JUnit with Mockito
 - Focus on service and business logic

```
import {render, screen} from '@testing-library/react'
import userEvent from '@testing-library/user-event'
import '@testing-library/jest-dom'
import Fetch from './fetch'

test('loads and displays greeting', async () => {
  // ARRANGE
  render(<Fetch url="/greeting" />)

  // ACT
  await userEvent.click(screen.getByText('Load Greeting'))
  await screen.findByRole('heading')

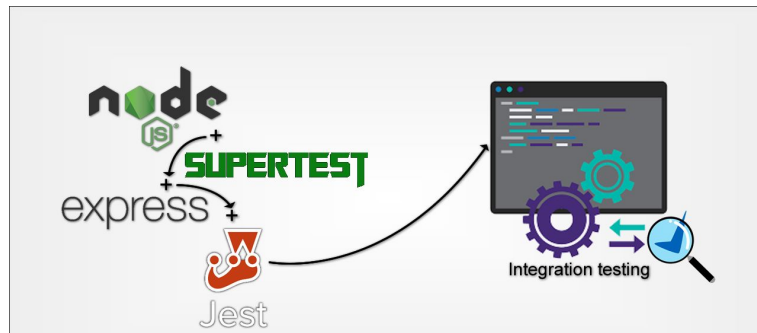
  // ASSERT
  expect(screen.getByRole('heading')).toHaveTextContent('hello there')
  expect(screen.getByRole('button')).toBeDisabled()
})
```



Integration Testing

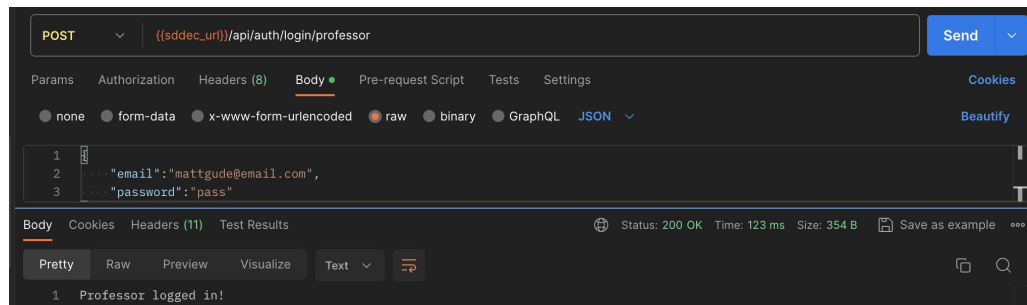
- **Frontend**

- API integration simulated with fetch-based test mocks.
- Planned E2E validation with Playwright



- **Backend**

- Postman for manual API validation
- JUnit + MockMvc for automated tests
- CI/CD pipeline runs tests on every push (GitLab)



Security Measures

Authentication & Access Control

- Passwords encrypted with BCrypt
- Role-based access controls (Student, TA, Instructor)

Data Protection

- CORS configured to block unauthorized origins
- Input validated via DTOs
- Only required data exposed in responses

Backend SafeGuards

- Secured with Spring Security filters
- API endpoints require valid token
- Permission checks on class/forum access

Classroom Testing

Classroom Testing

- Load Test - Test actual lecture setting
 - 70 concurrent users all uploading images.
 - Average incoming network traffic: 12.3MBit/s
 - Peak incoming network traffic: 220.31MBit/s
- UI Load Testing - Ensure website is fast & responsive
- Browser-Based Load Testing - Ensure website works on different browsers



Classroom Testing – Results

Ease Of Use

95% complemented SmartClass' ease of use.

Sign Up

95% had an easy sign-up experience.

Future Use

80% would prefer SmartClass over competitors in the future.

Intuitive

88% complemented how intuitive SmartClass was.

Anonymity

80% appreciated the anonymity features provided by SmartClass.

Feedback

- Improve Forum Design
- Mobile Design
- Self-Signed HTTPS

Feedback: Improvements

- Mobile design optimized.
- Need for text eliminated.

Future Plans With Feedback In Mind

- Move to AWS EC2.
- Get an actual HTTPS certification.

Conclusion

SmartClass

Interactive Learning Tool

Design

- React
- Spring Boot
- MySQL

Future

- Solve minor bugs
- Move to AWS
- Support other institutions



SmartClass