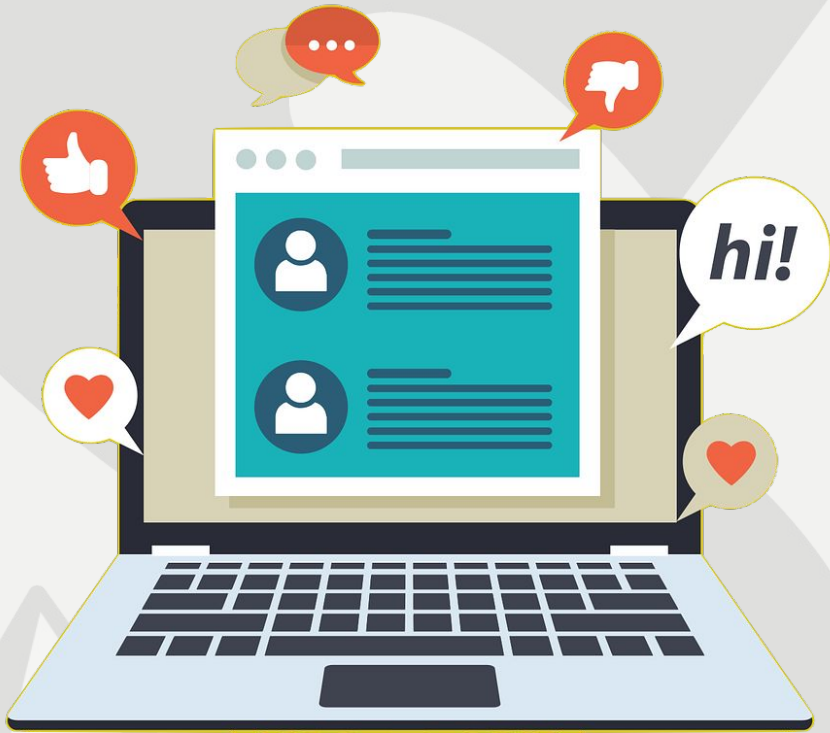


Sierra College Foundation

Byte-Sized

Sorea Asmai, Miles Brown, Quan Duong,
Denny Khuth, Matthew Mendoza,
Celso Quero, & Iulia Tarasova

Project Overview



Client

- Sierra College Foundation
- **Objective:** Foster a unified Sierra College alumni community

Problem

- Lack of a centralized platform for alumni interaction

Proposed Solution

- **Alumni Portal Development:** A dedicated website for alumni to:
 - **Connect:** Central hub for networking and updates.
 - **Contribute:** Streamlined donation process to support the Foundation.
 - **Collaborate:** Bulletin for job opportunities and event announcements
 - **Celebrate:** Calendar sharing for key events

Client Requirements

Foundation Support

- **Donation Functionality:** Secure & user-friendly donation system

Member Verification

- Verification process to authenticate alumni before granting site access

Networking & Opportunities

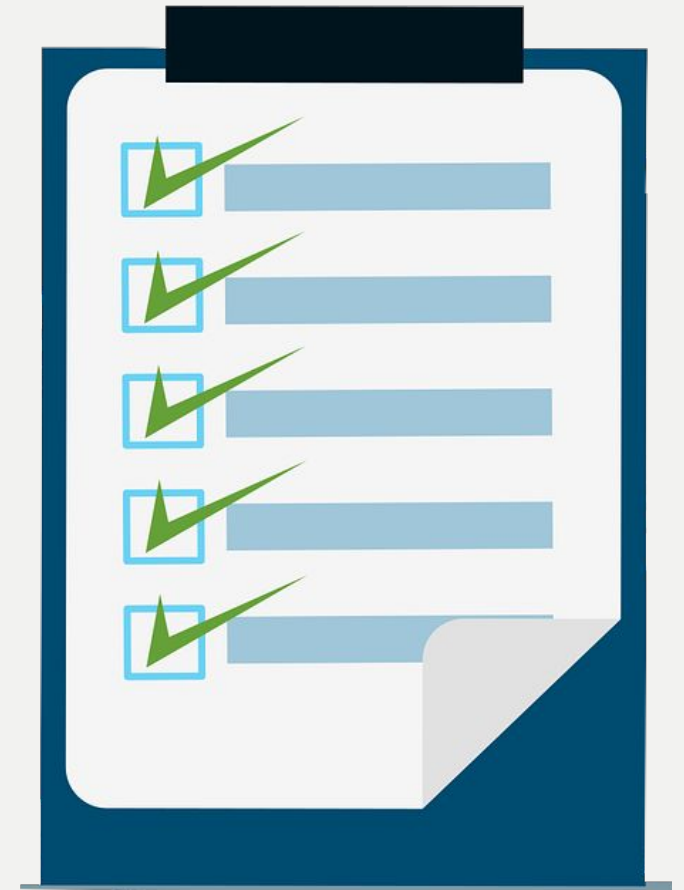
- **Job Board:** Platform for posting/browsing alumni job opportunities
- **Events Hub:** Location for posting/discovering alumni events

Event Calendar

- **Client-Editable:** An intuitive calendar interface for the client to update & share upcoming events

Alumni Directory

- **Search Capability:** Advanced search function to locate & connect with fellow alumni



Networking & Opportunities

JOB BOARD

Exclusive Alumni Job Board

- A dedicated platform for alumni to share job opportunities

Alumni-Driven Content

- Members of the alumni community can contribute job postings, fostering a collaborative environment

Administrative Oversight

- A user-friendly dashboard empowers administrators to manage and curate the content posted on the platform

EVENT BOARD

Alumni Event Hub

- A specialized platform exclusively for alumni to share & discover upcoming events

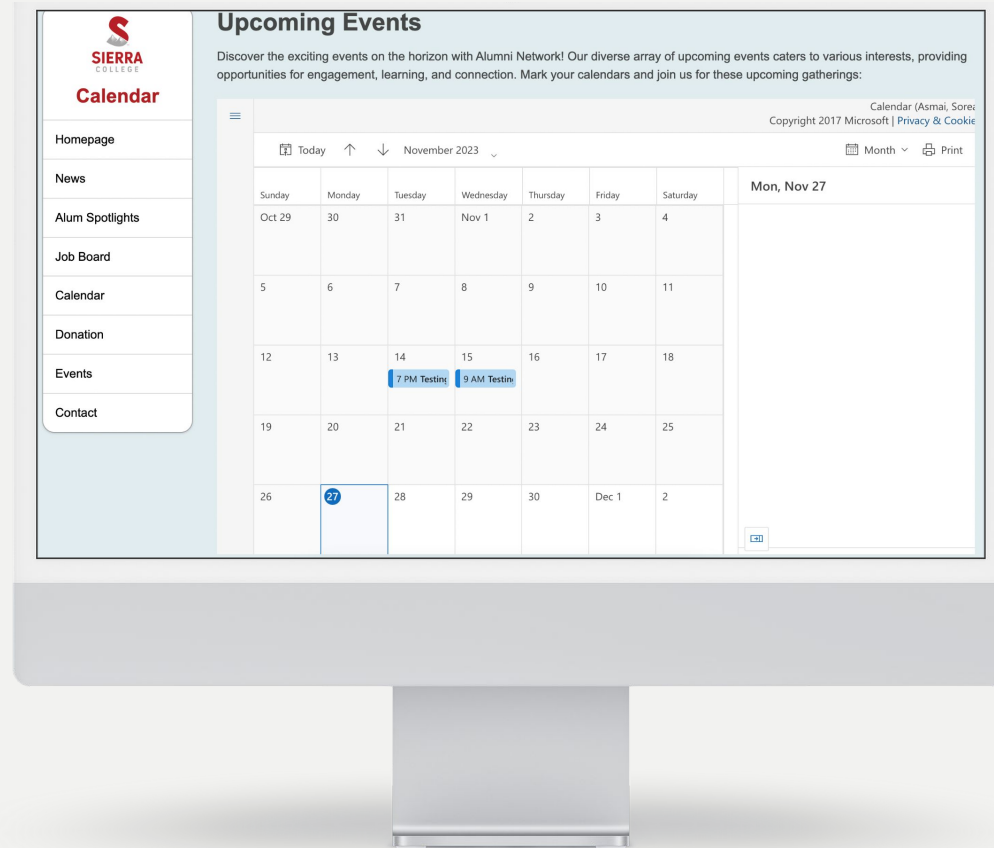
Alumni-Hosted Events

- Members of the alumni community can contribute & promote various events on the board

Administrative Control:

- An intuitive dashboard enables administrators to oversee & manage the events posted on the platform

Event Calendar



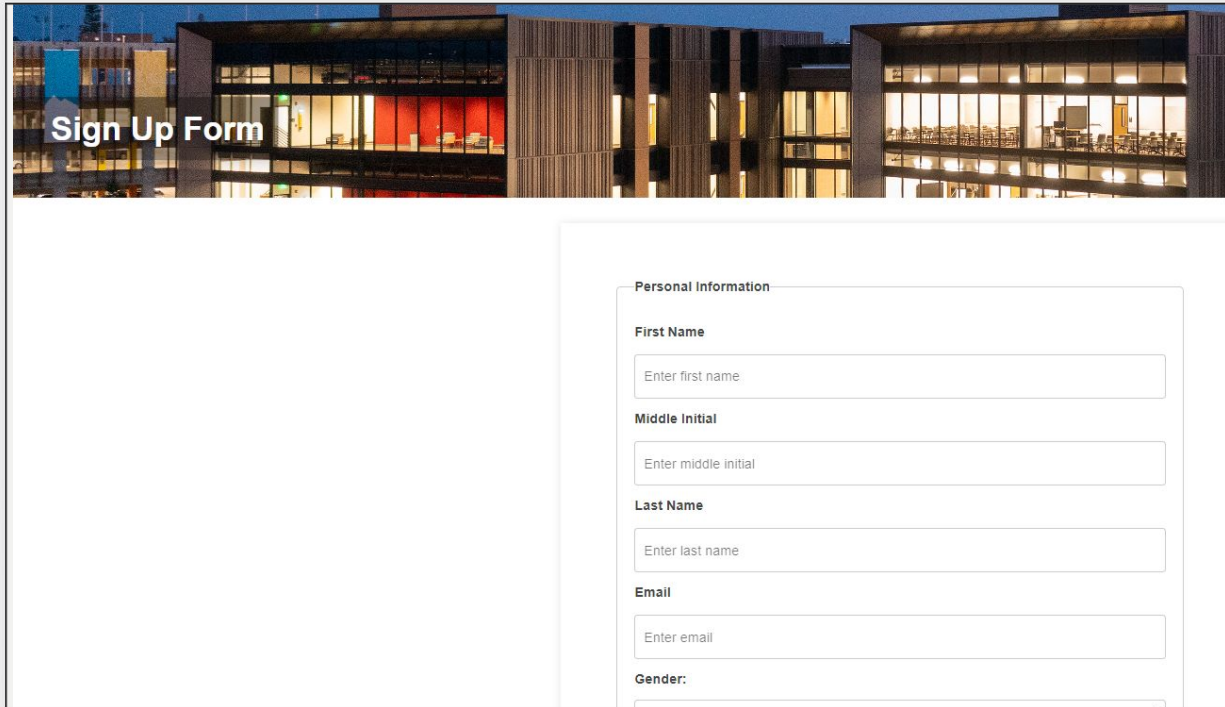
Outlook Calendar Integration

- Display key dates & college events seamlessly

Collaborative Editing

- Multiple users in the office can update calendar for real-time accuracy on site

Member Verification



Sign Up Form

Personal Information

First Name

Enter first name

Middle Initial

Enter middle initial

Last Name

Enter last name

Email

Enter email

Gender:

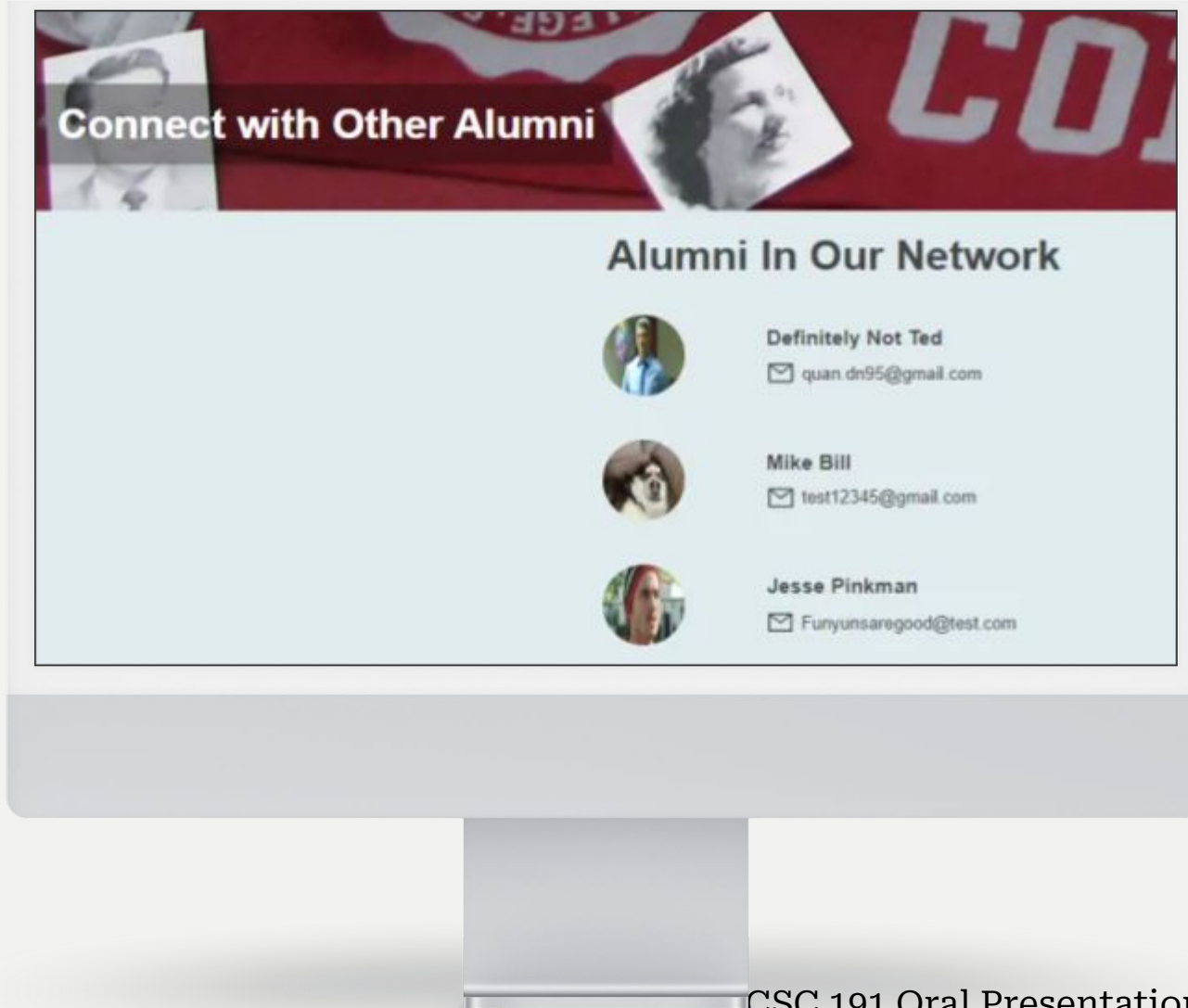
Multi-step Alumni Verification

- Straightforward process to sign up for the website that goes through admin approval

Alumni Feedback

- Opportunity to share their experience with Sierra College/learn about preferences for involvement

Alumni Directory



Signature Feature

- Available only for alumni who have been verified

Expand your Network

- Key information available in the directory, such as names and email addresses

Foundation Support

Current Presenter:
Iulia Tarasova

Donate Today

Gifts to Sierra College Foundation provide support to the programs, services, and students of Sierra College. Your gifts are tax-deductible to the extent allowed by law. Choose a fund to donate to:

Student Assistance Fund Endowment (SAFE): Tuition may be free, but what about living expenses? SAFE offers financial assistance to all students demonstrating a need to defray the cost of living.

Guardian Scholars Program for Former Foster Youth: Offers financial assistance to emancipated youth (former foster youth) to help them overcome barriers to success and realize college is an option.

Innovation: Join us in supporting Sierra College's innovative approaches to our region's challenging education and workforce issues.

Endow-a-Bed Program: A philanthropic investment of \$250,000 will create an endowment that offers free room and board, in perpetuity, to one student. Any contribution to support this program is helpful!

Basic Needs: Sierra College's new Basic Needs Center is a lifeline of resources for students facing hardships.

Veterans' Success Center: Help us support programs that benefit our veterans returning to civilian life.

Emergency Fund for Students: Assist students by providing emergency support that can help them through a critical time and one-time emergencies making it possible for them to stay in school.

Support New and Existing Scholarships: Support more than 150 existing scholarships available to Sierra College students or consider creating your own scholarship opportunity.

Donate

Multiple Donation Funds

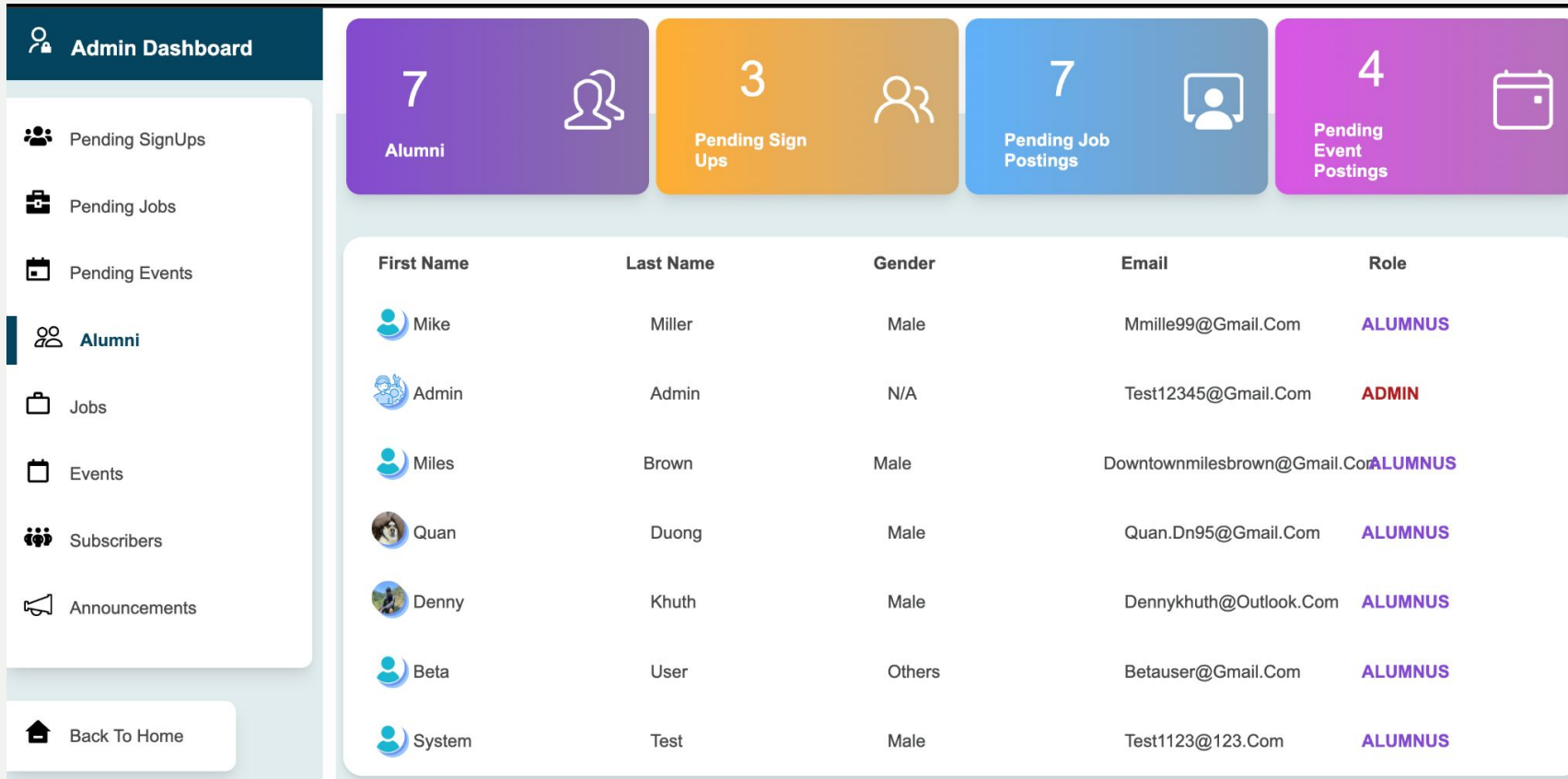
- Enabling donors to support causes that resonate with them

Raiser's Edge Integration

- Secure donation process with ability to track proceeds from website

Dashboard

Current Presenter:
Quan Duong



Main features:

- Accept/Deny sign-ups, job postings and event postings.
- View current alumni, active postings and subscribers.
- Send announcements to subscribers.
- Show useful stats that can be easily seen.

Design & Demo

- Show the design of the product
- Show each product feature
- Demo the Working/Deployed application

Burn Down Chart



Implementation

- Implemented scrum
 - 2 week sprints
 - used discord channels instead of daily meetings
 - assigned task at the beginning of the week in Jira.
 - code needs 2 reviews before merging into main



Tech stacks

- JavaScript, TypeScript & SCSS
- Node.js, Express
- MySQL
- VSCode
- Firebase, outlook

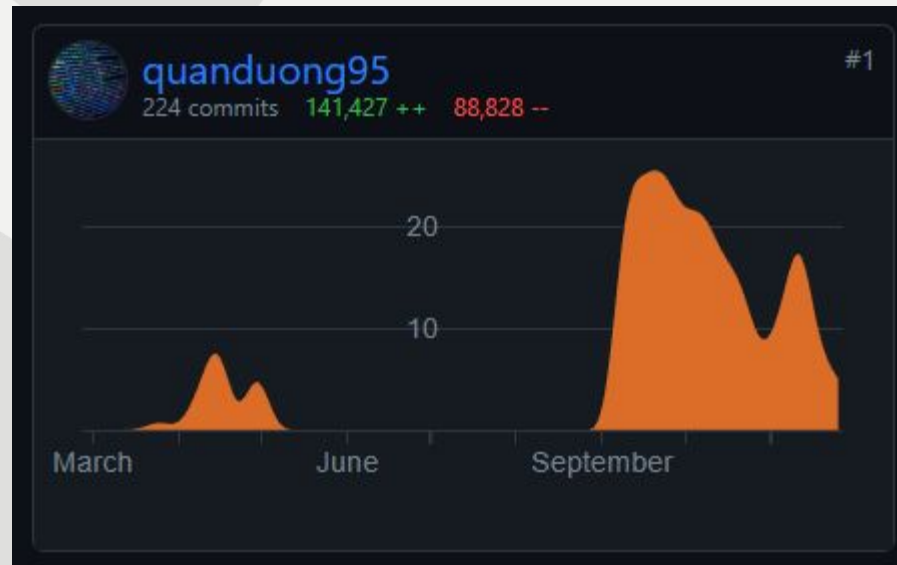


Firebase



Implementation

Team MVP

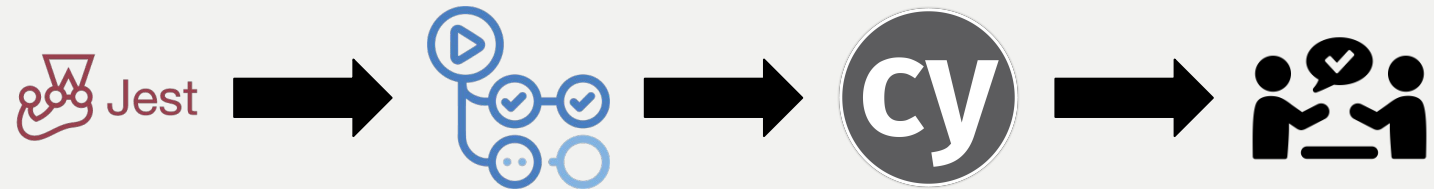


- Over 30,000+ lines of code
- 568 commits
- In the last month we added 14,684 lines of code

Testing

Functional test plan

- Unit testing
 - Jest
- Integration testing
 - GitHub Actions
- Systems testing
 - Cypress
- Acceptance testing
 - Is validated with client



Testing

Unit testing

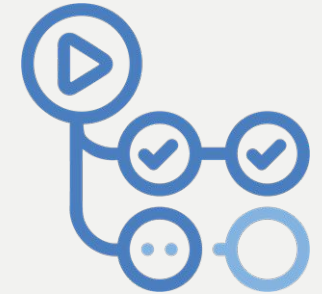


```
1 import { updateUserInfo, updateUserRole } from '../api/updateData';
2 import axios from 'axios';
3
4 jest.mock('axios');
5
6 describe('updateUserInfo', () => {
7   it('should update user information successfully', async () => {
8     const mockResponse = { data: { result: { success: true } } };
9     axios.put.mockResolvedValue(mockResponse);
10
11     const payload = {
12       name: 'John Doe',
13       email: 'john.doe@example.com',
14       phone: '123-456-7890',
15     };
16
17     const result = await updateUserInfo(payload);
18
19     expect(result).toEqual({ success: true });
20     expect(axios.put).toHaveBeenCalledWith(`${process.env.REACT_APP_BASE_URL}user-info`, payload);
21   });
22 });
```

```
1 describe('deletePendingSignUp', () => {
2   it('should delete a pending sign-up request successfully', async () => {
3     const mockData = { success: true };
4     axios.delete.mockResolvedValue({ data: { result: { data: mockData } } });
5
6     const result = await deletePendingSignUp('uid', 'id');
7     expect(result).toEqual(mockData);
8     expect(axios.delete).toHaveBeenCalledWith(expect.stringContaining('pending-user'), expect.any(Object));
9   });
10
11   it('should throw an error when request fails', async () => {
12     axios.delete.mockRejectedValue(new Error('Network error'));
13
14     await expect(deletePendingSignUp('uid', 'id')).rejects.toThrow('Network error');
15   });
16 });
```

Testing

Continuous integration testing: GitHub Actions



```
1 name: Node.js CI
2
3 on:
4   push:
5     branches: ["main"]
6   pull_request:
7     branches: ["main"]
8
9 defaults:
10   run:
11     working-directory: ./client/
12
13 jobs:
14   build:
15     runs-on: ubuntu-latest
16
17     strategy:
18       matrix:
19         node-version: [18.x]
20         # See supported Node.js release schedule at https://node.dev/en/versions/
21
22     steps:
23     - uses: actions/checkout@v3
24     - name: Use Node.js ${{ matrix.node-version }}
25       uses: actions/setup-node@v3
26       with:
27         node-version: ${{ matrix.node-version }}
28         cache: "npm"
29         cache-dependency-path: "./client/package-lock.json"
30
31     - name: Install dependencies
32      run: npm ci
33
34     - name: Run tests
35      run: npm test
36
37     - name: Run build
38      run: npm run build --if-present
```

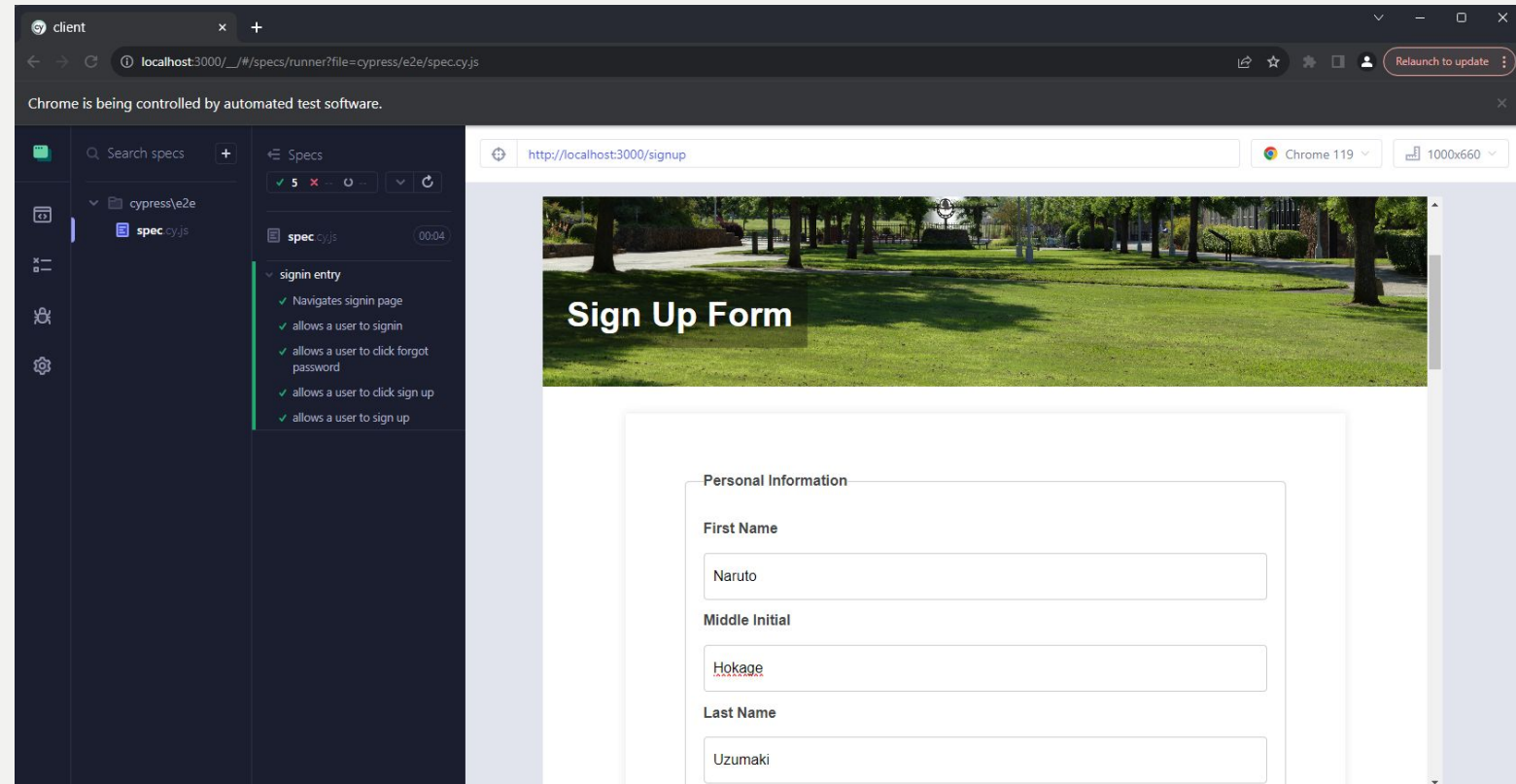
Testing

Integration/system testing: Cypress



```
D: > client
> npm run e2e
> client@0.1.0 e2e
> cypress open
```

```
JS spec.cy.js 9+, M X
cypress > e2e > JS spec.cy.js > describe("signin entry") callback
You, 29 seconds ago | 1 author (You)
1 describe("signin entry", () => {
2   it("Navigates signin page", () => {
3     cy.visit("http://localhost:3000/signin");
4   });
5   it("allows a user to signin", () => {
6     cy.visit("http://localhost:3000/signin");
7     cy.get("input[name=email]").type("test_user@tester.com");
8     cy.get("input[name=password]").type("password");
9     cy.contains("Sign in").click();
10  });
11
12  it("allows a user to click forgot password", () => {
13    cy.visit("http://localhost:3000/signin");
14    cy.contains("Forgot password?").click();
15  });
16
17  it("allows a user to click sign up", () => {
18    cy.visit("http://localhost:3000/forgot-password");
19    cy.contains("Sign Up").click();
20  });
21
22  it("allows a user to sign up", () => {
23    cy.visit("http://localhost:3000/signup");
24    cy.get("input[name=firstName]").type("Naruto");
25    cy.get("input[name=middleInitial]").type("Hokage");
26    cy.get("input[name=lastName]").type("Uzumaki");
27  });
28 });
29
```



Testing

Acceptance testing: Is validated with client

1. Client meeting is arranged
2. Demo implemented feature
3. Does the feature match client's expectations
4. Receive feedback from client

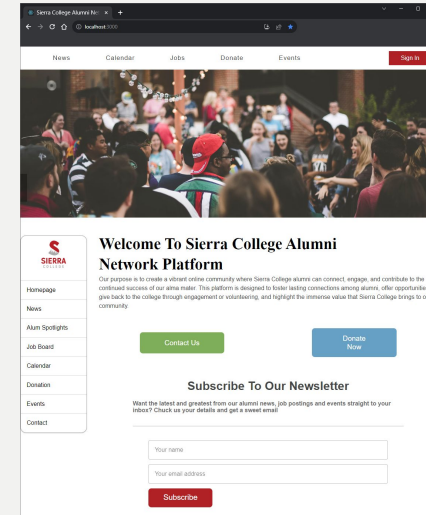


Testing

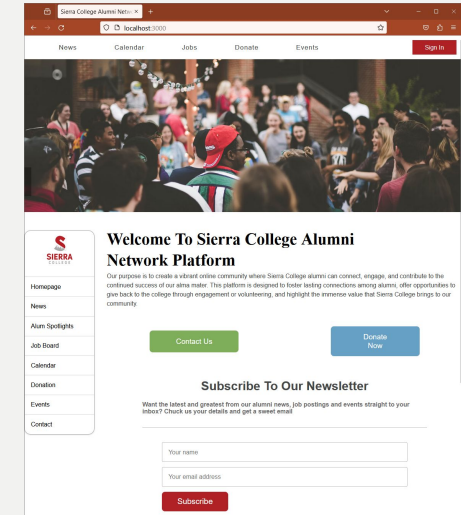
Non-functional testing

- Performance testing
- Usability testing
 - Feedback given by users
- Compatibility testing
 - Multiple browsers

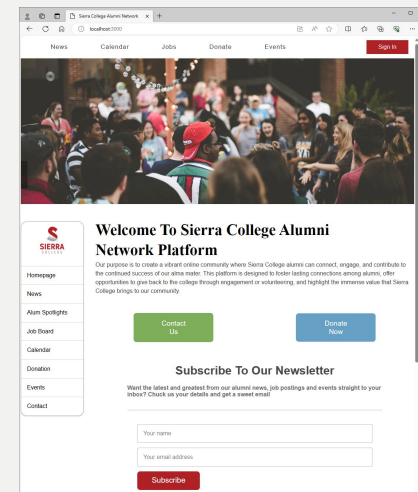
Current Presenter:
Matthew Mendoza



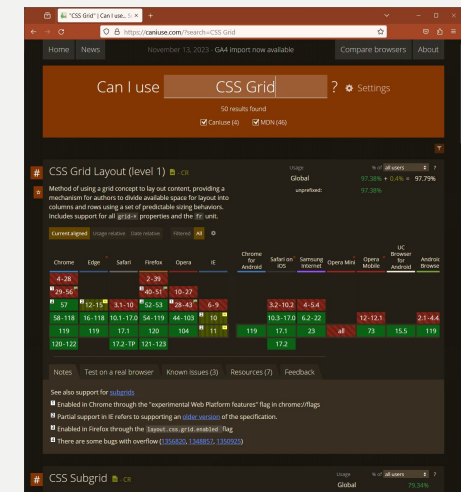
Chrome



Firefox



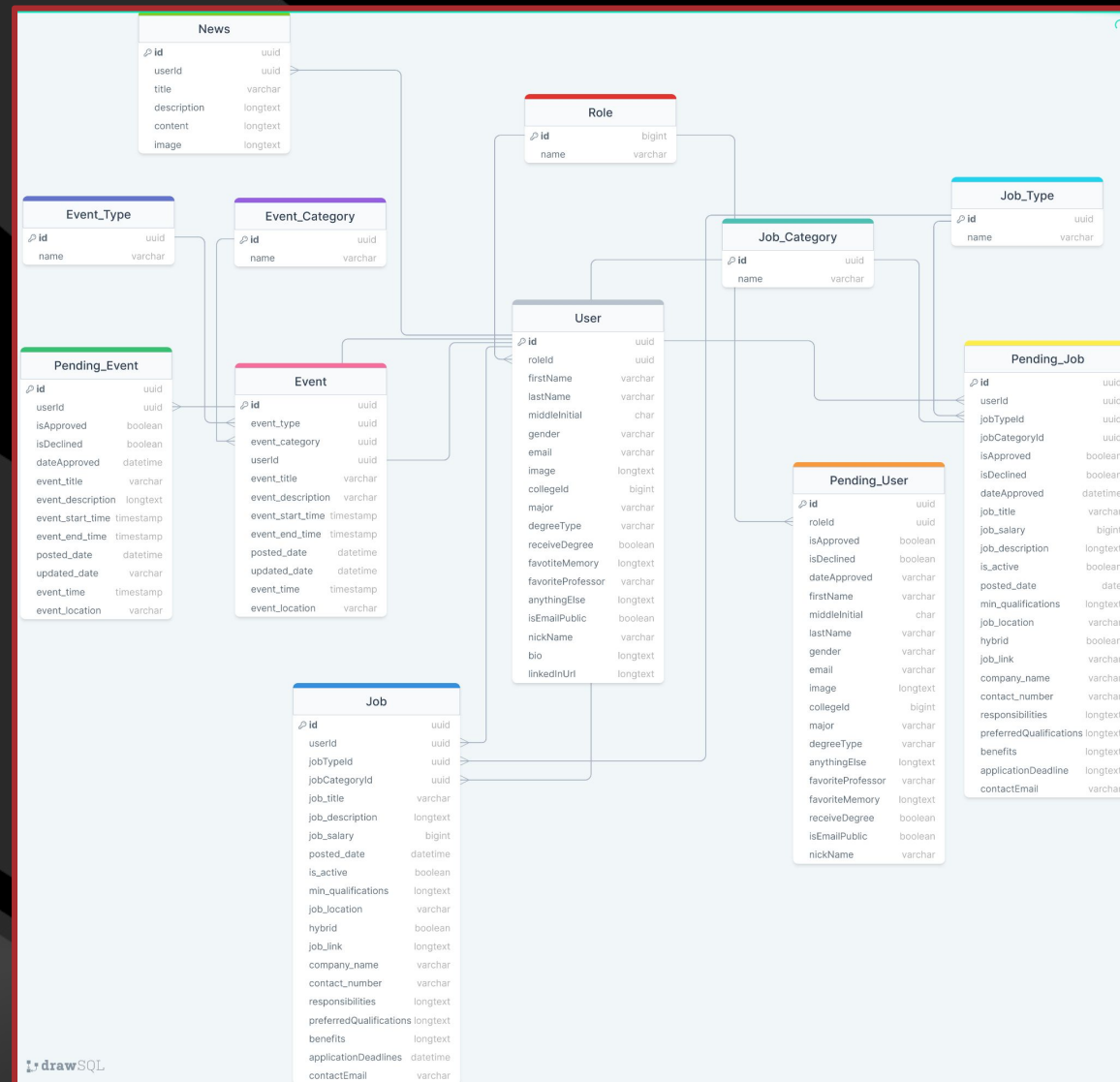
Edge



caniuse.com

Database Design

Current Presenter
Denny Khuth



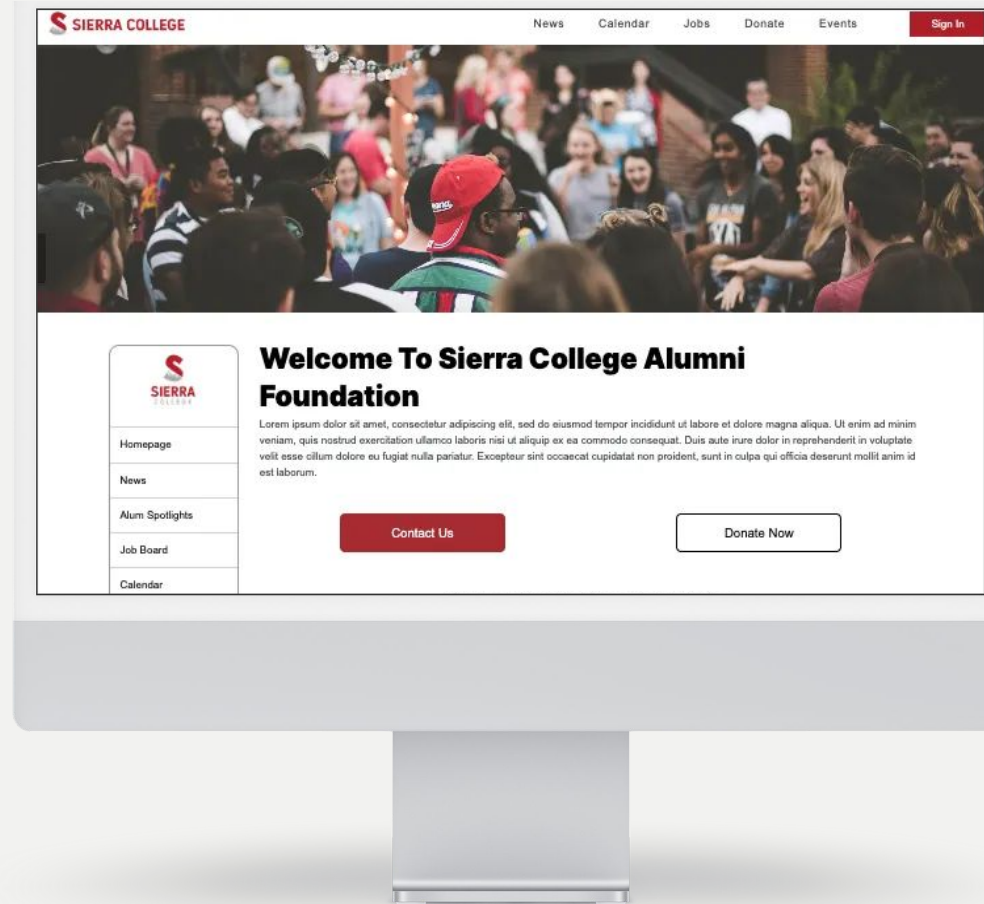
Deployment

- AWS EC2
 - Compute service for backend
 - Running on linux VM
- AWS S3
 - Serve static content (react-app)
- AWS RDS
 - Host MySQL database
 - Can use MySQL Workbench to interact with this DB.
- Cloudinary
 - Store Image files

Lesson Learned

Things Learned in Class

- **New Technologies**
 - AWS Cloud
 - React
 - Playwright
 - Jest
- **New Tools**
 - Effective Communication
 - Programming Improvement



Team Insights/Wisdom

- **Effective Communication**
 - Discord & Bi-Weekly Meetings
 - Timely Assistance & Improvement
- **Project Alignment**
 - Meeting Client's Exceptions
 - Consistent with JIRA

Conclusion

- **Client's feedback (so far):**
 - Positive response to website progress
 - Excited to receive final product
- **Area for Improvement:**
 - **Content Integration**
 - Include written content provided by client
 - **Trumba Access Integration**
 - Integration of Trumba and coordinate with the Marketing team for accreditation access
 - **New Section Added to Side Navigation Bar**
 - “Alum Spotlight”

Q&A

Thank you for attending!

Thank you for attending!