

# Class Ratings

Scarlet Fever

Huanli Gong, Alex Valentine, Sudi Yussuf,  
Sagar Patel, Saeema Sayed



# Implemented Features

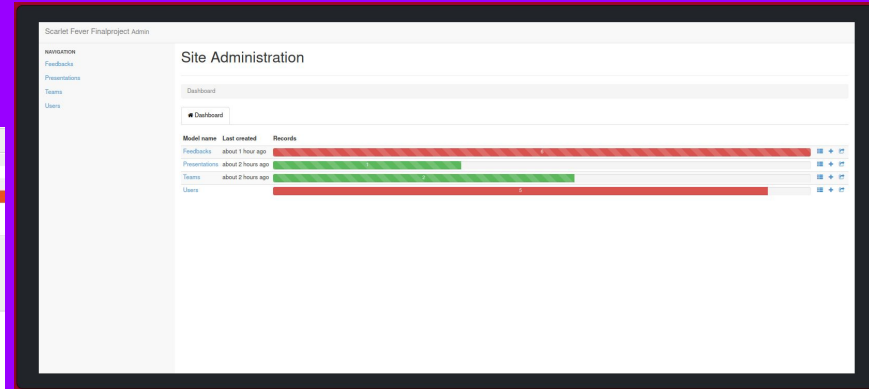
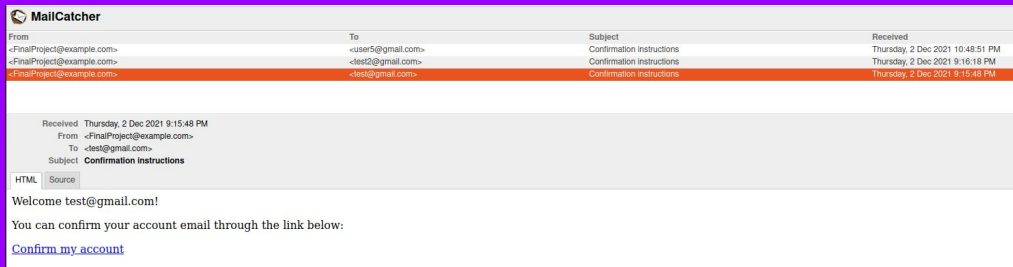
1. Instructor/TA can create presentation events for which evaluation can then be submitted
2. Audience members can submit scores and comments for a presentation
3. Administrative interface
4. Presenter has access to the feedback provided by the audience
5. Authentication/login
6. Javascript
7. Application looks professional

# Implemented Extensions

1. Prevent self-evaluation
2. Admin dashboard to monitor submissions and manage evaluations
3. Changing enrollment for students who add or drop the course
4. Different output formats like a .csv of all data

# Gems Used

- Gem 'devise'
- Gem 'cocoon'
- Gem 'rails\_admin'
- Gem 'mailcatcher'



# Authentication and Login

- Devise
  - Database\_authenticatable
  - Registerable
  - Recoverable
  - Rememberable
  - Validatable
  - Confirmable
- Username
  - Custom username column
  - Application controller
    - Adding new parameters
  - Changed `config.authentication_keys` to `:username`
  - Added username input fields to the views

# Authentication and Login

## Sign up

Username

First name

Last name

Email

Password

Password confirmation

Sign up

Sign in instead?

## Log in

Username

Password

☐ Remember me

Log in

Sign up instead?

Forgot your password?

Didn't receive confirmation instructions?

## Resend confirmation instructions

Email

Resend confirmation instructions

Sign in instead?

Sign up instead?

Forgot your password?

# Models

- User
  - belongs to and has many teams
  - has many feedbacks
  - directly integrated for use by Devise to authenticate logins
- Team
  - belongs to and has many users
  - has many presentations
- Presentation
  - belongs to a team
  - has many feedbacks
- Feedback
  - belongs to a user and a presentation

# Views and Controllers

- Models and Authentication
  - Admin
  - Application
  - Confirmation
  - Feedback
  - Home
  - Presentation
  - Teams
  - Users
- Views
  - CRUD
  - Routes to various pages
- Layout with shared navigation bar for consistency



# Layout and Front-end

- Bootstrap
  - Framework for website that provided CSS and Javascript compiled files
- Rails Admin
  - Robust backend management dashboard that allows the admin user to easily manage the entire web application
- Cocoon
  - Handles nest forms and attributed in one form with ease

# Prototype Demo



# Difficulties

- Experienced difficulty in adding Foundation so instead we went with bootstrap
- We ran into issues with the user model
- We haven't gotten to add Heroku
- Handled issues caused by adding nested attributes

# Future Extensions

- Switch to PostgreSQL and then deploy the application to Heroku
- Class roster
  - Allow the professor to select which students are part of the class
- Change the styling of the evaluation form
- Change the layout of the feedback page
  - Make it more accessible to view many different feedbacks at once

**Thank You!**