CSCE 606: Shockwave Kickers

Team members: Colton Simpson, Jacob Kelly, Kunal Vudathu, Rahul Shah, Stuart Nelson

Iteration 2

- **Team roles**: The team should elect a scrum master and product owner.
 - We have an agreed upon rotation that will shift with each iteration
 - Scrum Master: Kunal Vudathu
 - Product Manager: Stuart Nelson
- **Customer meeting date/time/place**: The team product owner should contact the customer and set up a regular weekly meeting.
 - Every Friday at 2pm in PETR
- SCRUM meeting date/time/place:
 - Every Tuesday and Thursday at 10am
- Github: https://github.com/Lightfoot-Heavy-Machinery/shockwave-kickers
- Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2595510
- Slack Workspace name:
 - Workspace: tamu.slack.com
 - o Channel Name: csce606 shockwave kickers
- Heroku Deployment: https://shockwave-kickers.herokuapp.com/courses

Work Accomplished:

- Migrated DB schema to what it should look like, had to change all controllers to work to what is expected
- Implemented complete authentication system
 - users can sign up, get email confirmations, reset passwords emails.
 - We also implemented the need to confirm email before signing in.
 - All the pages will be accessible only when the user has signed in.
 - A user can access only the information that they created, everything is private.
- Parse csv uploaded from howdy and creates student and course object to what it's needed
- The course page showcases all the students signed up in that course.
- Migrated all react code to html/bootstrap, this includes all pages like courses, students and home page
 - Accomplished it to being very close to figma diagrams below
- All this works in heroku as well!!

User stories selected in this iteration.

- Update db schema on backend and perform changes/migrations as needed
 - Migrate the db in heroku and local to finalized db schema
 - Update student and course controllers and UI components to be compatible with the new db schema
 - Show students under the same course in the same page
- Backend: Implement db update on json received by parse csv function
 - We are now able to parse the CSV roster file that is loaded from
 - Automatically builds Course and Student records, and write these records to the database
 - If a student is in a course that doesn't exist in the database, then the system will automatically create a new Course record before saving the Student

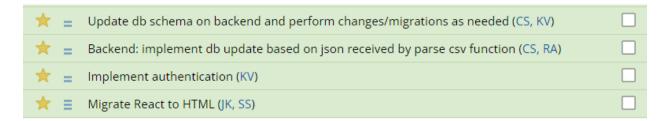
Implement Authentication

- Users can sign up, get email confirmations, reset passwords emails.
- We also implemented the need to confirm email before signing in.
- All the pages will be accessible only when the user has signed in.
- A user can access only the information that they created, everything is private.

Migrate React to html and build pages to what figma diagram looks like

- Build a base template for all the pages
- Courses page, student profile page, course profile page were built and stitched with backend and authentication
- Edit student and edit course were created to what it looks like figma

Stories in Pivotal Tracker:



Stories Added/Changed:

- For Iteration 3, We added the following stories:
 - Based on request from stakeholder, we added a story for Authentication done via SSO
 - A story to stories images via S3 bucket

Iteration 3 work:

- Get the upload feature working end to end, with it updating correctly on the new DB schema
- Basic functionality for Quiz feature
- Upload pictures and store in S3 bucket
- Get higher test coverage using rspec
- Get CI/CD pipeline working

Test Coverage Report

Unit Cases Coverage:

We increased coverage coverage from 78% to 91.26% as well as increased number of tests from 10 to 28 tests, this includes more edge case coverage as well as tests on authentication as well:

All Files (91.26% covered at 1.51 hits/line)

10 files in total.

103 relevant lines, 94 lines covered and 9 lines missed. (91.26%)

					Search:	
File	\$ % covered Å	Lines 🔷	Relevant Lines 🔷	Lines covered 🔷	Lines missed 🖣	Avg. Hits / Line 🏺
q app/controllers/students_controller.rb	90.00 %	77	40	36	4	1.53
q app/controllers/upload_controller.rb	90.00 %	26	10	9	1	1.80
\P app/controllers/courses_controller.rb	90.91 %	84	44	40	4	1.55
9 app/controllers/home_controller.rb	100.00 %	5	3	3	0	1.00
q app/helpers/courses_helper.rb	100.00 %	2	1	1	0	1.00
q app/helpers/home_helper.rb	100.00 %	2	1	1	0	1.00
q app/helpers/students_helper.rb	100.00 %	2	1	1	0	1.00
q app/helpers/upload_helper.rb	100.00 %	2	1	1	0	1.00
Q app/models/course.rb	100.00 %	2	1	1	0	1.00
9 app/models/student.rb	100.00 %	2	1	1	0	1.00

Showing 1 to 10 of 10 entries