Links

Github: https://github.com/nishant-basu-tamu3/student_knowledge_system
Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2721032
Deployed Application: https://hive-sks-c9da748e2491.herokuapp.com/

Dates

The Sprint lasted from October 21st to November 3rd.

Team Members

Total Points Completed: 20
Scrum Master: Ellika Mishra, completed 2
Product Owner: Prakhar Suryawansh, completed 2
Dev: Nishant Basu, completed 6
Dev: Raunak Kharbanda, completed 4
Dev: Susheel, completed 4
Dev: Mukil , completed 4
Dev: Aum Palande, completed 4

Sprint Goal

Our Sprint goal was to make students searchable by any field, fix the upload script and notes section for notes to be viewed/edited by the professor.

Sprint Achievements

We completed most of our sprint backlog, primarily updating the upload functionality to match the new Howdy data. We also completed the feature that allowed the professor to search students by any given fields or tags. Other features included adding a notes section for the professor to add notes for students individually. We faced an issue with the edge case when a student name had ',' and the application did not work as expected and also a server error on quiz refresh page. We continued to work on both rubocop and code coverage, reducing the rubocop offenses to below 100 and increasing the coverage to 90%. Overall, we achieved 20 out of the 20 story points planned. We have implemented the search functionality, course filtering and tag filtering functionality. One of the things left to be done is making the notes for individual students editable visible, some issues persisted with this feature.

Sprint Backlog

- Notes for individual students pop-ups not visible
- Crashing of application when parsing student names with ','
- Quiz server error on refresh occasionally
- Code coverage > 90%

Burn Down:

Points accepted vs. points prioritized
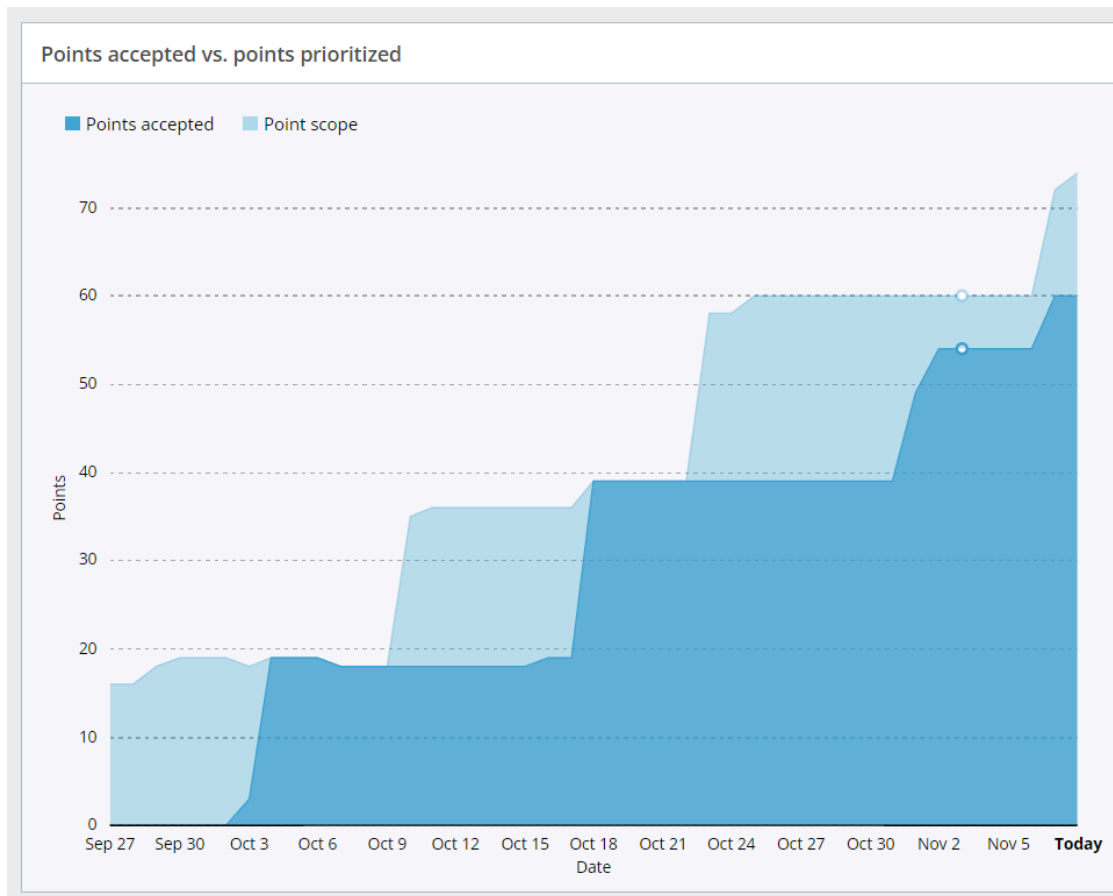
- ★ Students Searchable by any field (name, UIN, email) (MU, SU) [#188415842]
  2pts
- ★ Add filter for current student/previous student (NI) [#188471788]
  2pts
- ★ Filter the quiz by courses (NI) [#188471781]
  2pts
- ★ code coverage (NI, AU, MU, EL) [#188471815]
  2pts
- ★ make tag not mandatory (AU) [#188471777]
  1pt
- ★ Fix Dropdowns for Create Student (AU) [#188483140]
  2pts
- ★ Fix default values and refactor/complete upload script (AU) [#188471797]
  2pts

Finished Stories (2 stories | 4 points)

- ★ Fix double commas on the name imports (RA) [#188471774]
  2pts
- ★ create note pop-up for professor, give option to edit note (PS, RA) [#188471745]
  2pts

Started Stories (1 story | 2 points)

- ★ fix the update student button to post (EL) [#188471775]
  2pts

Design Diagram

Old students page (without note pop-up feature):

New students page (with editable note pop-up feature):

Course(s)/Semest

Tags:

Filter Students List

Yearbook View

×

Good Student!

Save Changes

Amdtha Prasad, .   amdthap@email.tamu.edu   CSCE000 - Fall 2024   None - Delete this student
                                                                                                              View/Edit Notes

Quiz Feature:

Student Knowledge System   Home   Courses   Students   Upload   Settings ⌄

Quiz Setup

Double-click the options to add them to the text areas below.

Course: Semester: Section:

| CSCE000 | Fall 2024 | 111 |

CSCE000   Fall 2024   111

Start the Quiz
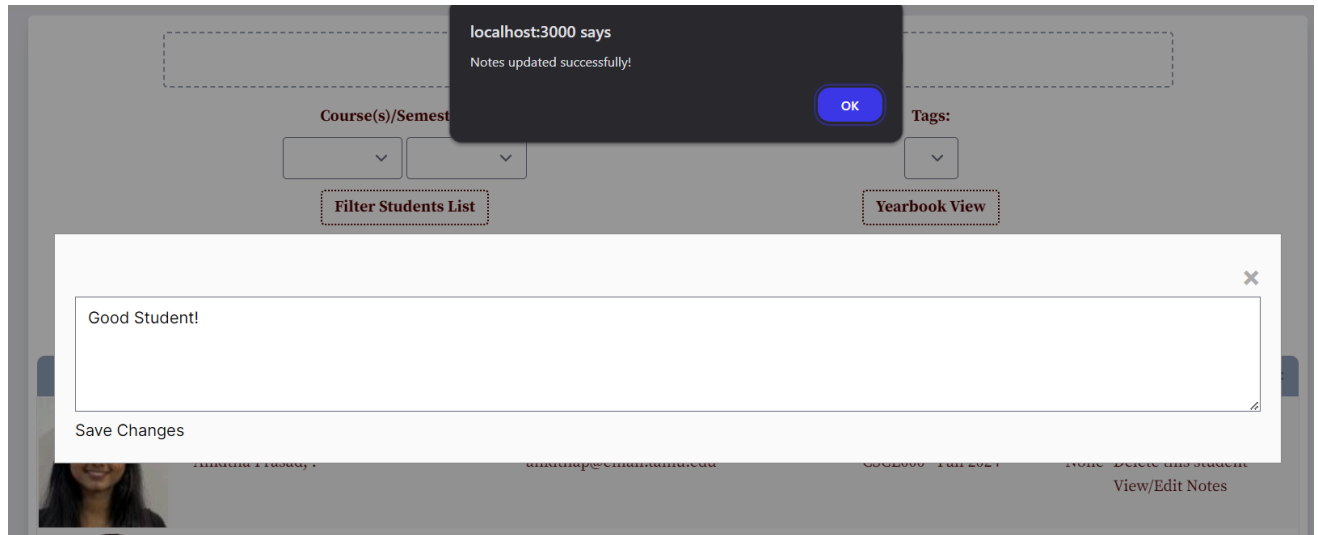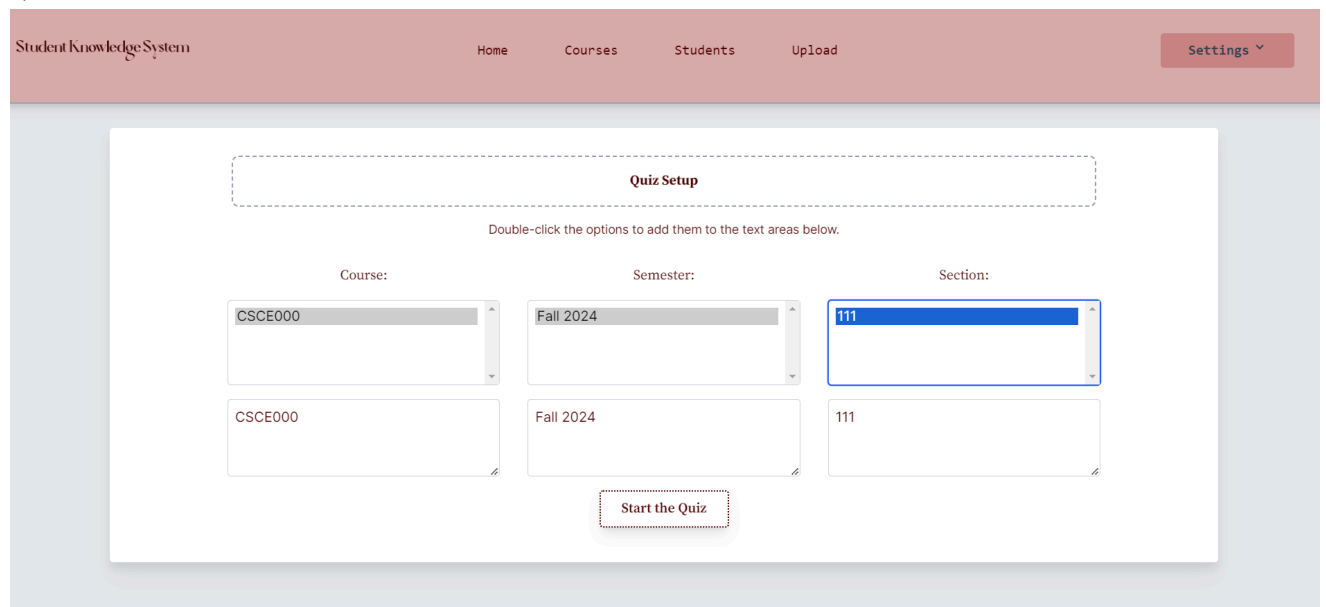
Who is this student?

- ○ Likhith Majjigapu
- ○ Raunak Kharbanda
- ○ Kushal Nitin Lahoti
- ○ Yash Sameer Phatak
- ○ Phani Jyothi Kurada
- ○ Riddhi Prakash Ghate
- ○ Parth Tusham
- ○ Pavithra Gopalakrishnan

Submit

Student Knowledge System   Home   Courses   Students   Upload   **Settings ⌄**

Correct Answer!

Try Another

Student Knowledge System   Home   Courses   Students   Upload   **Settings ⌄**

Incorrect Answer.

The correct student was Sahithi Duppati.

Try Another

😞

Old:

New:

Documentation of change :

Docs HIVE SKS

Code Evaluation

Code Climate : https://codeclimate.com/github/nishant-basu-tamu3/student_knowledge_system

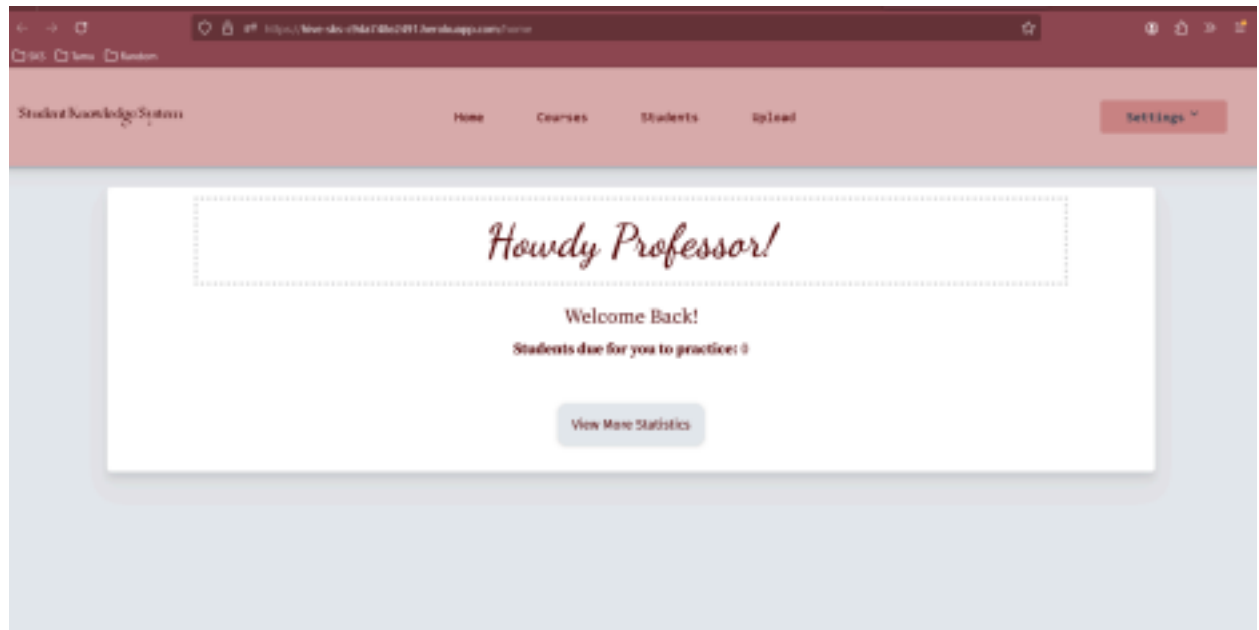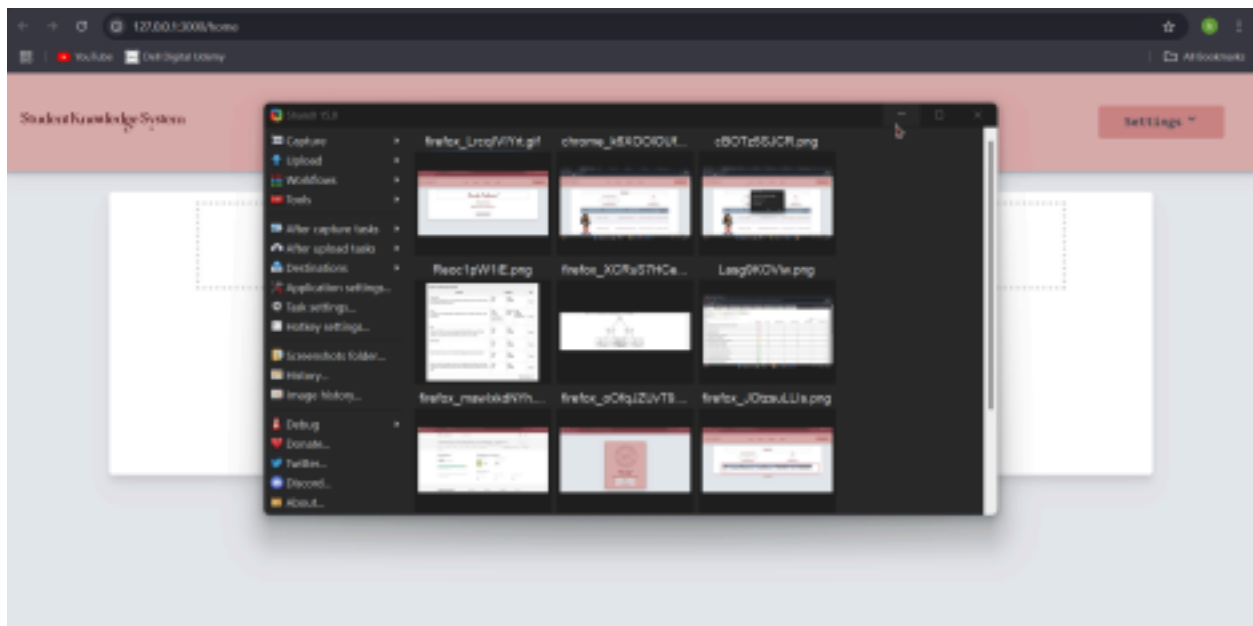**CODE CLIMATE**   Product ∨   Pricing   Developers   **Sign up**   Login

nishant-basu-tamu3/student_knowledge_system ⌂

Overview   Progress   Issues   Code   Filters   Trends

Last ⎇ master build 8 days ago   ↻ Refresh

☆ Star

Breakdown

166 FILES

MAINTAINABILITY

TEST COVERAGE

Codebase summary

MAINTAINABILITY

A   7 hrs

TEST COVERAGE

☂

Repository stats

CODE SMELLS

3

DUPLICATION

0

OTHER ISSUES

0

Simple Cov : 90.14 %

All Files (90.14%) Controllers (90.76%) Channels (100.0%) Models (81.82%) Mailers (100.0%) Helpers (92.86%) Jobs (100.0%) Libraries (100.0%)   Generated 7 days ago

All Files (90.14% covered at 226.13 hits/line)

31 files in total.
578 relevant lines, 521 lines covered and 57 lines missed. (90.14%)

Search:

File	% covered ▲	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
⚲ app/controllers/users/omniauth_callbacks_controller.rb	0.00 %	54	4	0	4	0.00
⚲ app/models/student.rb	66.67 %	43	18	12	6	21.78
⚲ app/controllers/student_courses_controller.rb	83.33 %	42	24	20	4	1.21
⚲ app/models/course.rb	83.33 %	61	36	30	6	5.67
⚲ app/controllers/students_controller.rb	87.36 %	302	182	159	23	25.54
⚲ app/helpers/courses_helper.rb	88.89 %	18	9	8	1	2.67
⚲ app/controllers/courses_controller.rb	89.74 %	203	117	105	12	13.70
⚲ app/controllers/upload_controller.rb	98.59 %	149	71	70	1	1658.46
⚲ app/channels/application_cable/channel.rb	100.00 %	6	2	2	0	1.00
⚲ app/channels/application_cable/connection.rb	100.00 %	6	2	2	0	1.00
⚲ app/controllers/application_controller.rb	100.00 %	33	14	14	0	222.93
⚲ app/controllers/courses/history_controller.rb	100.00 %	16	6	6	0	2.17
⚲ app/controllers/home_controller.rb	100.00 %	48	29	29	0	31.90
⚲ app/controllers/sessions_controller.rb	100.00 %	30	13	13	0	3.08
⚲ app/controllers/static_controller.rb	100.00 %	10	4	4	0	2.00
⚲ app/controllers/users_controller.rb	100.00 %	26	12	12	0	11.42

RSpec examples: 84 - 26 new examples.
Cucumber scenarios: 28 - Same number as before sprint.

Customer Meeting

We met with Dr. Lightfoot on November 6, from 4:30pm to 5:30pm. He was satisfied with the search functionality, customized quizzing functionality and tag creation feature. He mentioned the server error on quiz refresh and some additional features he would like to include.

BDD/TDD:

1. Create a quiz flier to quiz students on customized semesters only. This needed adding some UI checks and backend functionality with no need for additional tests.

2. Rubocop offenses less than 100

    The code remains intact in this part just the way it was written is changed hence it does not change the existing functionality, hence no TDD or BDD can be done on this type of story.

3. Adding search functionality on name, UIN or any other field for students was a logic based check no need for additional tests.

4. Make tag not mandatory.
5. Fix dropdown for create students.
6. Fix default values and refactor/complete upload script
7. Delete Course button
8. Scrape Images from html file

    The three stories above encompass the changes to the upload controller, which involve modifying the existing cucumber and rspec tests.

    a. BDD: ./app/features/upload.feature

```
Scenario: Upload correctly
    When I create account and sign in as "team_cluck_admin2@gmail.com"
    Given I am on the upload page
    When I upload a csv file
    When I upload a htm file
    And I input form information
    When I click save
    Then I should see the upload was successful
```

    b. TDD: ./app/spec/controllers/upload_controller_spec.rb

```
context 'with valid file and contents' do
  before do
    @course1 = Course.create(course_name: 'test course1', teacher: @user.email, section: '008',
                              semester: 'Fall 2002')
  end

  it 'creates student entries after uploading a valid csv and .htm file' do
    csv = fixture_file_upload('export.csv', 'text/csv')
    complete_webpage_file = fixture_file_upload('Howdy Dashboard _ Howdy.htm', 'text/html')
    params = { csv_file: csv, complete_webpage_file:, course_temp: @course.course_name, section_temp: @course.section,
               semester_temp: @course.semester }
    post :parse, params:
    susheel = Student.find_by firstname: 'Susheel'
    expect(susheel.firstname).to eq('Susheel')
    expect(flash[:notice]).to eq('Upload successful!')
  end

  it 'updates students that already exist' do
    @prev_student = Student.create(firstname: 'Susheel', lastname: 'VadaAkekaruppath', sin: '236002222', email: 'zcb@tamu.edu',
                                    classification: 'G7', major: 'CPSC', teacher: @user.email)
    csv = fixture_file_upload('export.csv', 'text/csv')
    complete_webpage_file = fixture_file_upload('Howdy Dashboard _ Howdy.htm', 'text/html')
    params = { csv_file: csv, complete_webpage_file:, course_temp: @course1.course_name, section_temp: @course1.section,
               semester_temp: @course1.semester }
    post :parse, params:
    uploaded_student = Student.find by sin: '236002222'
    puts "ups: #{uploaded_student.inspect}"
    expect(@prev_student.email).not_to eq(uploaded_student.email)
    expect(flash[:notice]).to eq('Upload successful!')
  end
end
```

9. Delete Button for students and courses table
   a. BDD: ./app/features/student.feature:

```
Scenario: Delete a student
    When I sign in as "team_cluck_admin@gmail.com"
    And I go to the students page
    When I click "New student"
    When I fill in student "firstname" with "New"
    When I fill in student "lastname" with "Student"
    When I fill in student "email" with "newstudent@email.com"
    And I click "Create Student"
    Then I should see "Student was successfully created"
    And I go to the students page
    And I click the delete button for student "New"
    Then I should see "New student"
```

10. Note pop up

   TDD: ./app/spec/controllers/students_controller_spec.rb

```
it 'retrieves notes successfully' do
  get :notes, params: { id: @student.id }
  expect(response).to have_http_status(:success)
  expect(JSON.parse(response.body)).to eq({ 'notes' => @student.notes })
end

it 'updates notes successfully' do
  new_notes = 'Updated Notes!'
  patch :update_notes, params: { id: @student.id, notes: new_notes }
  expect(response).to have_http_status(:success)
  expect(JSON.parse(response.body)).to eq({ 'success' => true })
  @student.reload
  expect(@student.notes).to eq(new_notes)
end

it 'handles notes update failure' do
  # Assuming your Student model validates presence of notes
  @student.update(notes: nil) # or whatever validation you have
  patch :update_notes, params: { id: @student.id, notes: nil }
  expect(response).to have_http_status(:success)
end
```

10. Code coverage > 90%
    To add more coverage, we added test cases for existing controllers to cover uncovered lines and added a test file for the course model.
11. Route user to be on homepage in active session
    Ensuring the user is not logged out when visiting the root page does not require any additional testing. No buttons or link tabs were changed that would cause cucumber tests to fail.

    Note: All the cucumber and rspec tests that were failing after the modification to the upload controller were corrected. For brevity, only the ones pertaining to the controller are shown above, however all tests where a view was modified to use the new image format or data needed to be uploaded were updated to ensure they were passing.