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 14th to October 18th.

Team Members

Total Points Completed: 21
Scrum Master: Susheel Vadakkekuruppath, completed 7
Product Owner: Mukil Senthilkumar, completed 7
Dev: Nishant Basu, completed 8
Dev: Raunak Kharbanda, completed 5
Dev: Ellika Mishra, completed 2
Dev: Prakhar Suryavansh, completed 5
Dev: Aum Palande, completed 6

Sprint Goal

Our Sprint goal was to update the upload functionality of the application to match the new Howdy website. This will allow the application to remain functional once the Howdy website is changed.

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 was left over from the previous sprint. The other features worked on were routing the user to be on the homepage while logged in, and adding a delete button to clear the database. 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 21 out of the 21 story points planned. We have implemented the delete functionality on the students page, allowing individual students to be removed directly, along with the show functionality that displays student details. This feature can be customized according to the client's requirements. However, the option to delete all students enrolled in a particular course is not yet implemented, but can be added if the client requests it.

Sprint Backlog

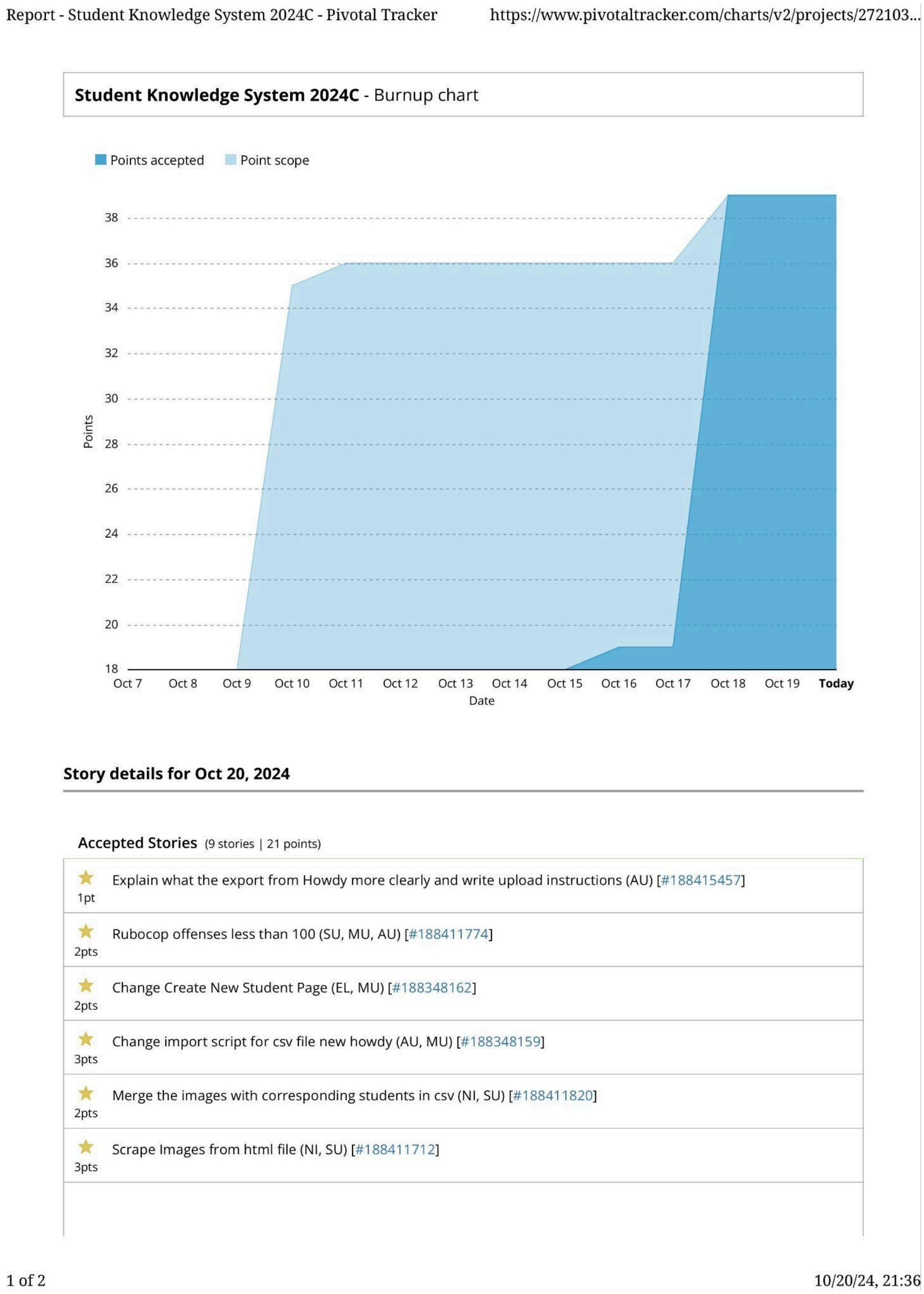
- Explain the export from Howdy more clearly and write upload instructions

- Rubocop offenses less than 100
- Change Create New Student Page
- Change import script for csv file new howdy
- Scrape Images from html file
- Merge the images with corresponding students in csv
- Delete Button for students and courses table
- Code coverage > 90%

+ Route user to be on homepage in active session

  Previously, when users logged in and were redirected to the home page, if they went to the root page, they were logged out and shown the login page. In order to preserve the session status till the user manually logs out, we made sure the user remains logged in by redirecting the root page to the home page.

Burn Down:



Report - Student Knowledge System 2024C - Pivotal Tracker https://www.pivotaltracker.com/charts/v2/projects/272103...

Student Knowledge System 2024C - Burnup chart

Story details for Oct 20, 2024

Accepted Stories (9 stories | 21 points)

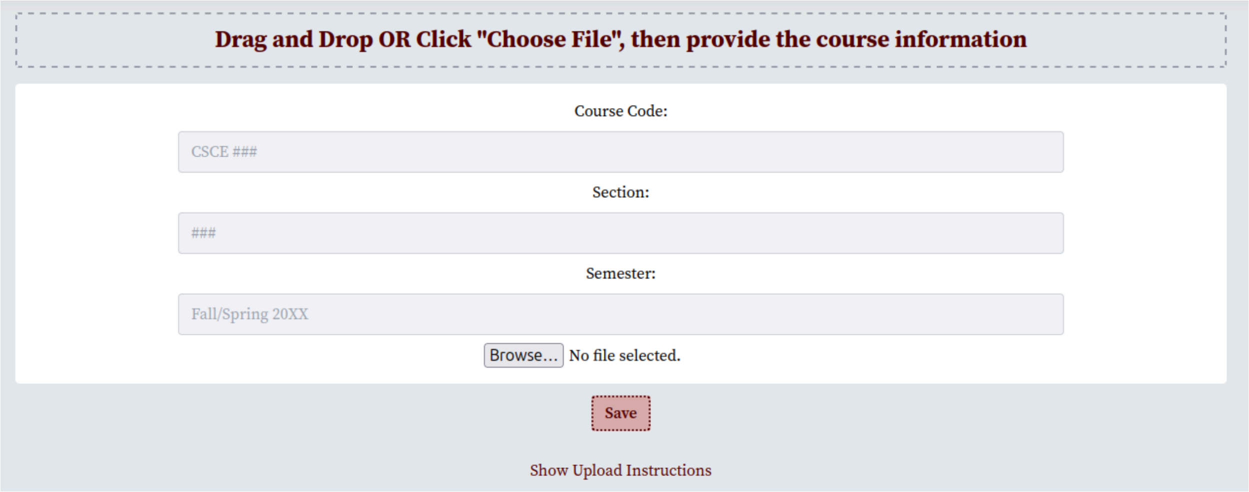
★ 1pt	Explain what the export from Howdy more clearly and write upload instructions (AU) [#188415457]
★ 2pts	Rubocop offenses less than 100 (SU, MU, AU) [#188411774]
★ 2pts	Change Create New Student Page (EL, MU) [#188348162]
★ 3pts	Change import script for csv file new howdy (AU, MU) [#188348159]
★ 2pts	Merge the images with corresponding students in csv (NI, SU) [#188411820]
★ 3pts	Scrape Images from html file (NI, SU) [#188411712]

1 of 2 10/20/24, 21:36

★ 3pts	Route user to be on homepage in active session (NI) [#188411671]
★ 3pts	Delete Button for students and courses table (RA, PS) [#188411590]
★ 2pts	Code coverage > 90% (PS, RA) [#188411778]

Design Diagram

Old upload page:



Drag and Drop OR Click "Choose File", then provide the course information

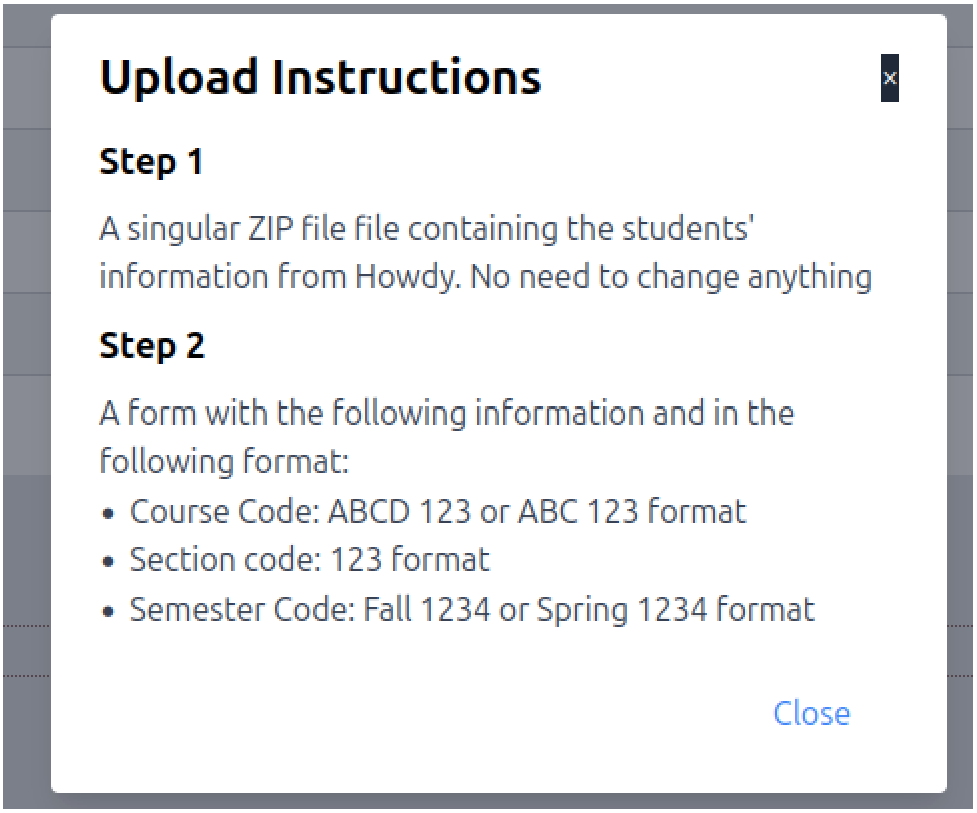
Course Code: CSCE ###

Section: ###

Semester: Fall/Spring 20XX

No file selected.

Old upload instructions:



Upload Instructions

Step 1

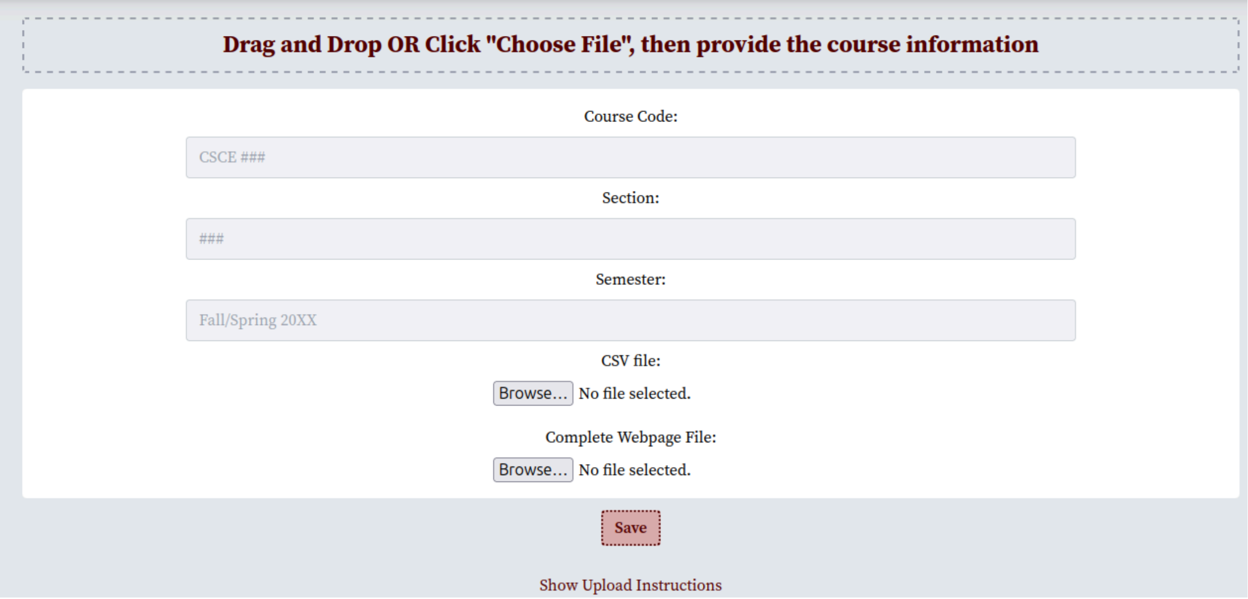
A singular ZIP file file containing the students' information from Howdy. No need to change anything

Step 2

A form with the following information and in the following format:

- Course Code: ABCD 123 or ABC 123 format
- Section code: 123 format
- Semester Code: Fall 1234 or Spring 1234 format

New upload page:



Drag and Drop OR Click "Choose File", then provide the course information

Course Code: CSCE ###

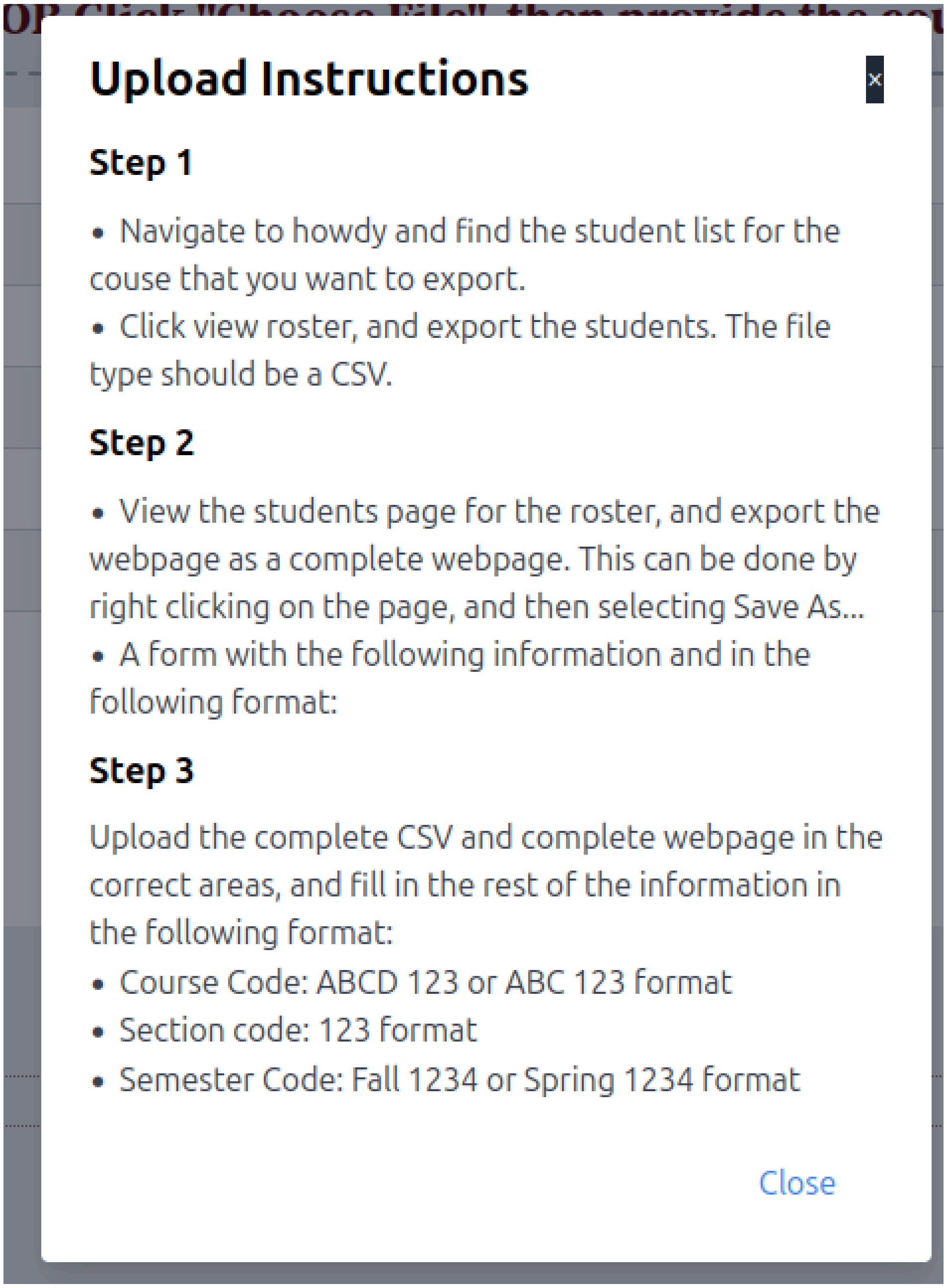
Section: ###

Semester: Fall/Spring 20XX

CSV file: No file selected.

Complete Webpage File: No file selected.

New upload instructions:



Upload Instructions

Step 1

- Navigate to howdy and find the student list for the couse that you want to export.
- Click view roster, and export the students. The file type should be a CSV.

Step 2

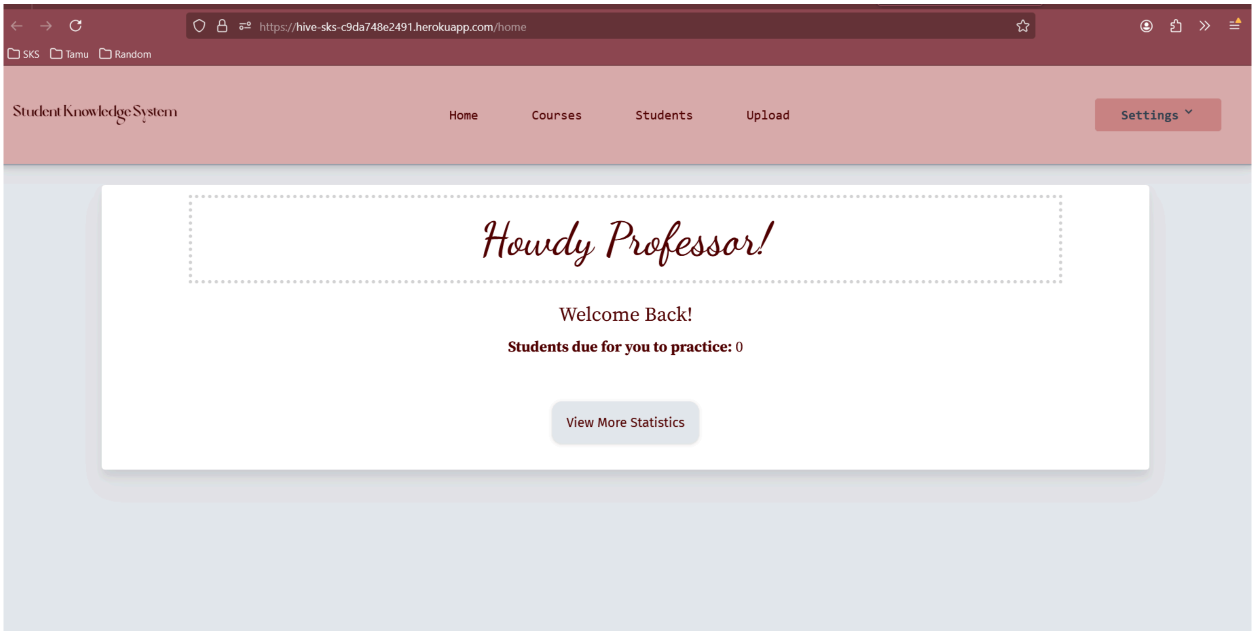
- View the students page for the roster, and export the webpage as a complete webpage. This can be done by right clicking on the page, and then selecting Save As...
- A form with the following information and in the following format:

Step 3

Upload the complete CSV and complete webpage in the correct areas, and fill in the rest of the information in the following format:

- Course Code: ABCD 123 or ABC 123 format
- Section code: 123 format
- Semester Code: Fall 1234 or Spring 1234 format

Route user to home page if logged in:
Old:

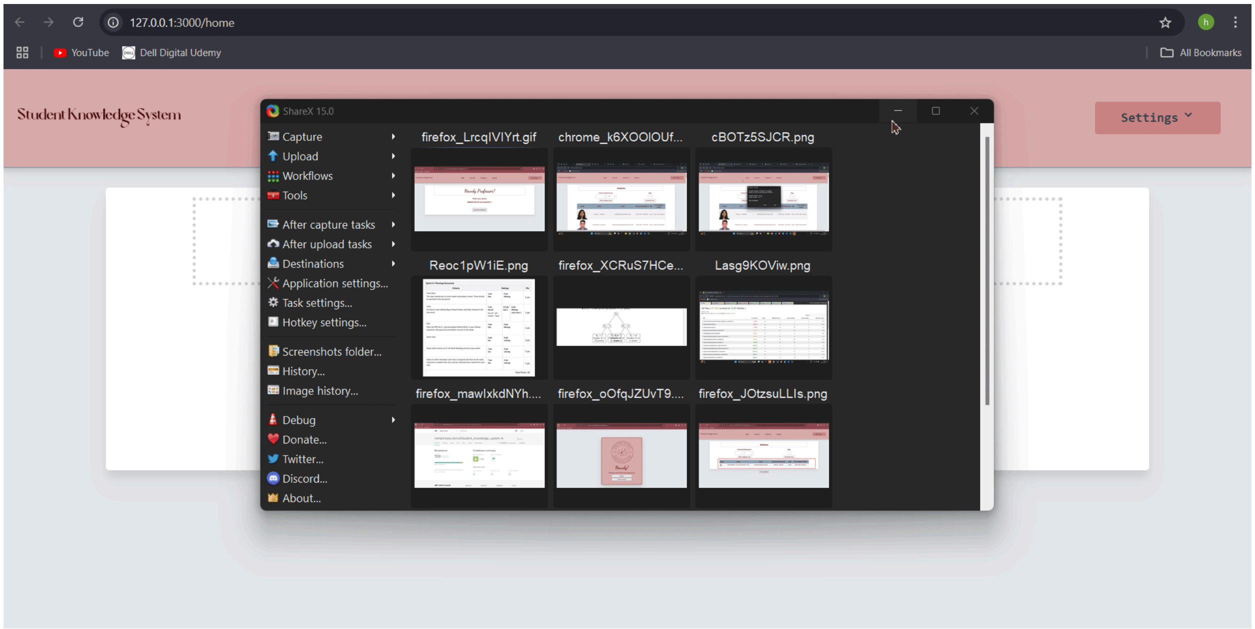


Howdy Professor!

Welcome Back!

Students due for you to practice: 0

New:

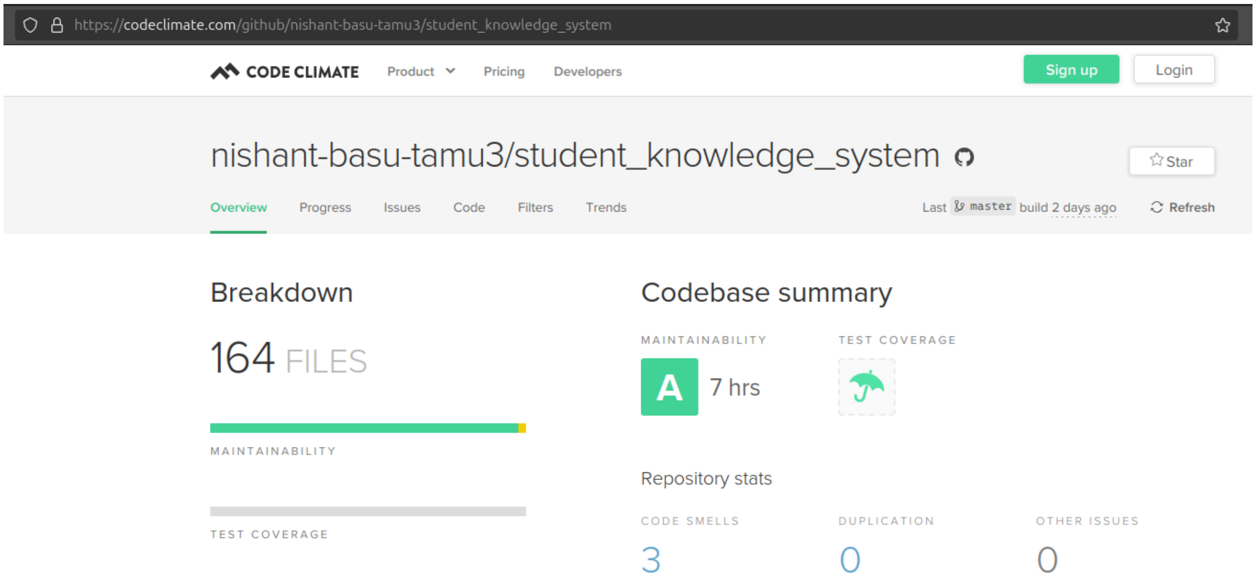


Documentation of change :

Docs HIVE SKS

Code Evaluation

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



nishant-basu-tamu3/student_knowledge_system

Last master build 2 days ago

Breakdown

164 FILES

MAINTAINABILITY

TEST COVERAGE

Codebase summary

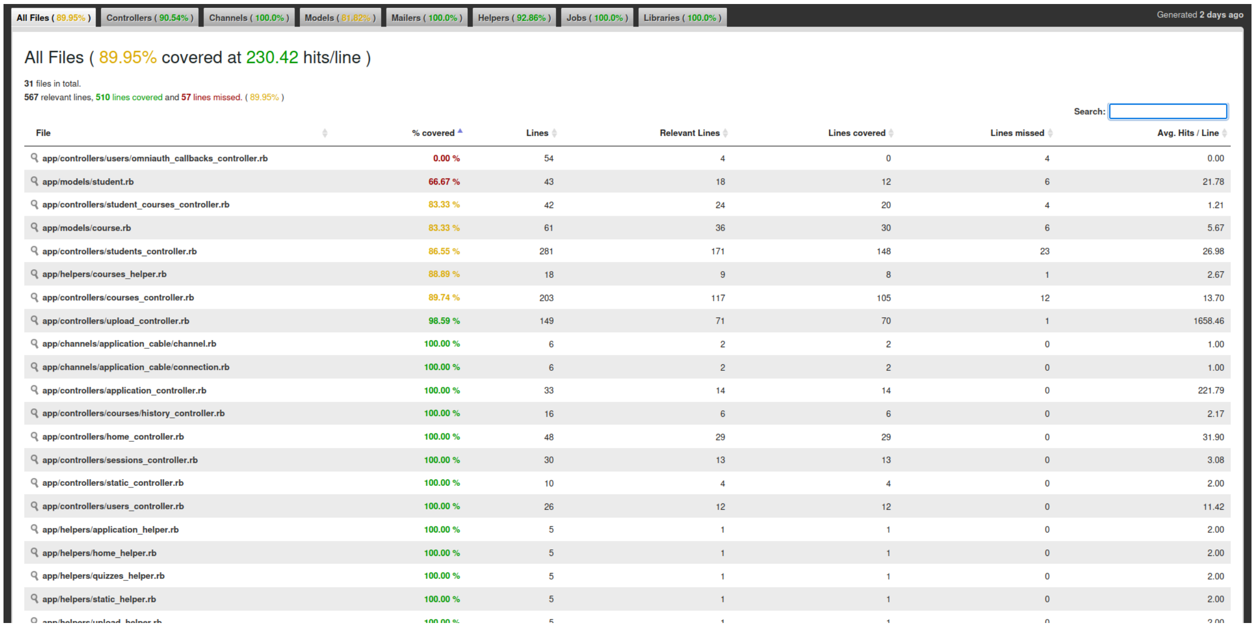
MAINTAINABILITY: A 7 hrs

TEST COVERAGE

Repository stats

CODE SMELLS	DUPLICATION	OTHER ISSUES
3	0	0

Simple Cov : 89.95 %



All Files (89.95% covered at 230.42 hits/line)

31 files in total.
567 relevant lines, 510 lines covered and 57 lines missed. (89.95%)

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	86.55 %	281	171	148	23	26.98
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	221.79
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
app/helpers/application_helper.rb	100.00 %	5	1	1	0	2.00
app/helpers/home_helper.rb	100.00 %	5	1	1	0	2.00
app/helpers/quizzes_helper.rb	100.00 %	5	1	1	0	2.00
app/helpers/static_helper.rb	100.00 %	5	1	1	0	2.00
app/helpers/upload_helper.rb	100.00 %	5	1	1	0	2.00

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

Customer Meeting

We met with Dr. Lightfoot on October 22nd, from 4:30pm to 5:30pm. He was satisfied with the updated implementation of the upload feature, and gave suggestions for improvements in other features of the app that will be included in the subsequent sprints.

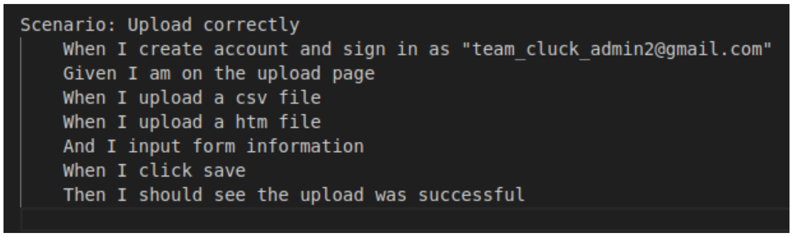
BDD/TDD:

1. Explain the export from Howdy more clearly and write upload instructions
   Updating the upload instructions was a modification of an existing feature, and did not require additional testing.

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.

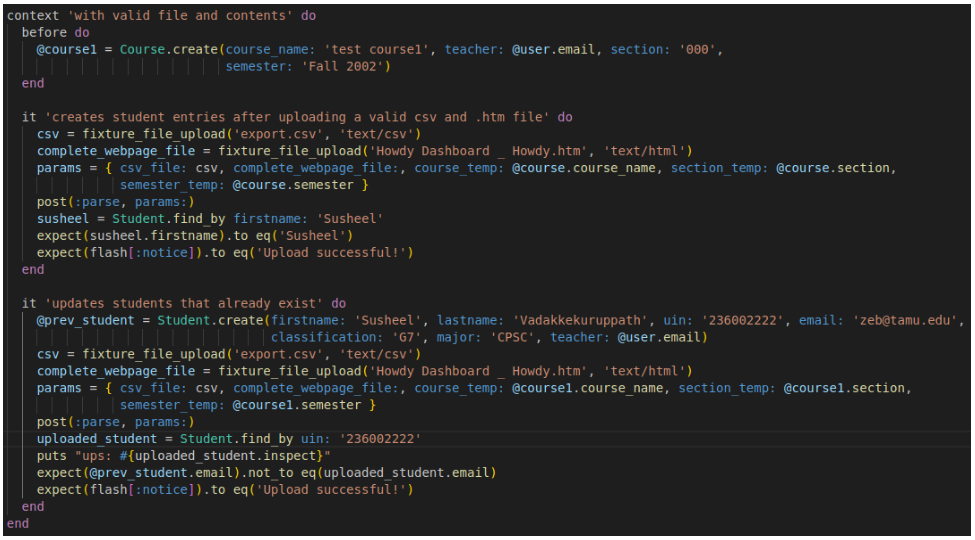
3. Change Create New Student Page
   Changing the create new student page to include drop downs for existing majors and classifications was purely a UI change and therefore does not require any additional testing. No buttons or link tabs were changed that would cause cucumber tests to fail.

4. Change import script for csv file new howdy
5. Scrape Images from html file
6. 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

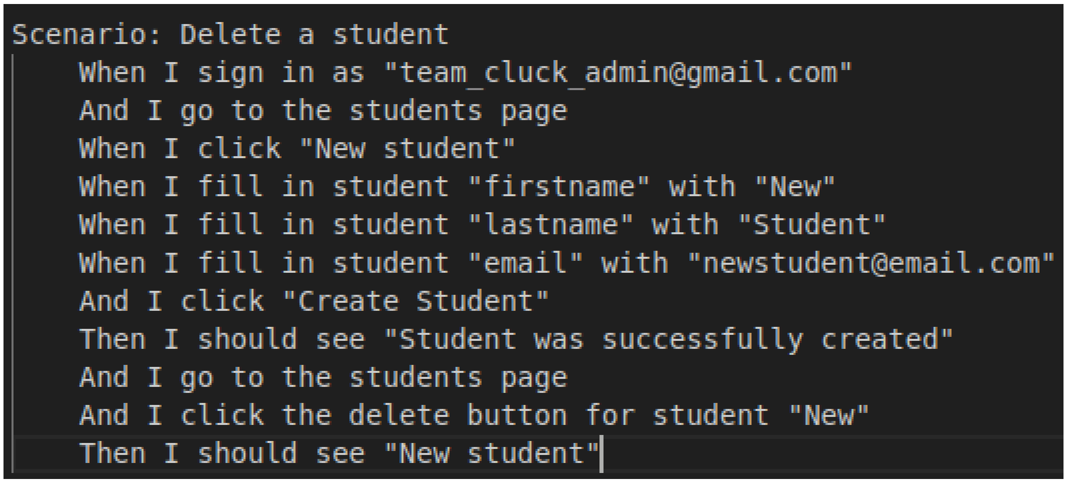


```ruby
context 'with valid file and contents' do
  before do
    @course1 = Course.create(course_name: 'test course1', teacher: @user.email, section: '000',
                             semester: 'Fall 2002')
  end

  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: 'Vadakkekuruppath', uin: '236002222', email: 'zeb@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 uin: '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
```

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



```gherkin
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"
```

8. 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.
9. 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.