Student Performance Tracker

Technical Work Summary

Contributors

Client: Mark Floryan

Developers:

Jimmy Patterson

Corey Lando

Taylor Nelson

Andrew Abraham

Connor Anderson

Jack Herd

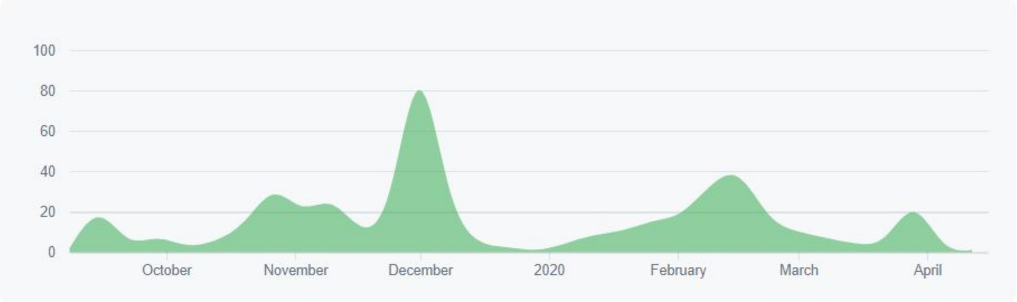
Alex Nguyen

Ryan Kann

High-Level Design

The Student Performance Tracker is an

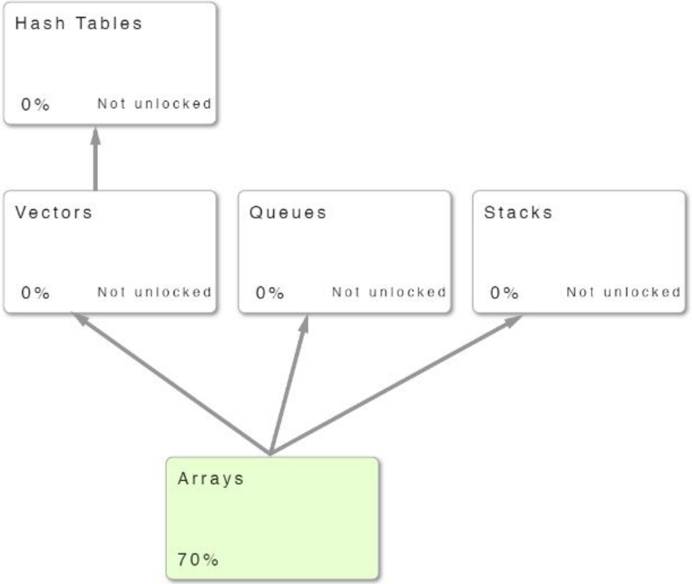
- Online Course Environment
  - Taking (and retaking) Quizzes
  - Reviewing and Administering Grades



High-Level Design

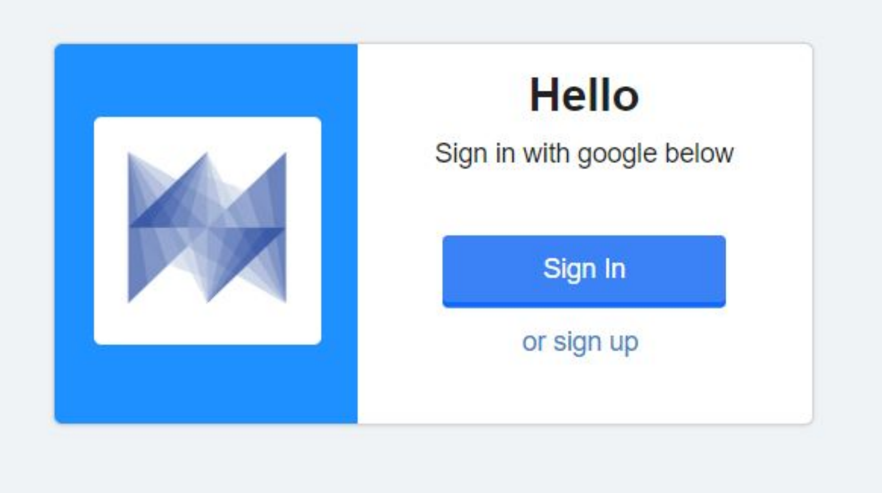
The Student Performance Tracker is a

- Specialized Pedagogical Tool
    - Graph-Based Structure
    - Competency-Bucket Grading
    - Adjustable Grading Metrics



Fall ‘19 and Spring ‘20 Goals

- Security
    - Improving Google OAuth efficiency
    - Protecting exposed API Endpoints
- Course Alignment
    - Changing grading policies
    - Locking course topics

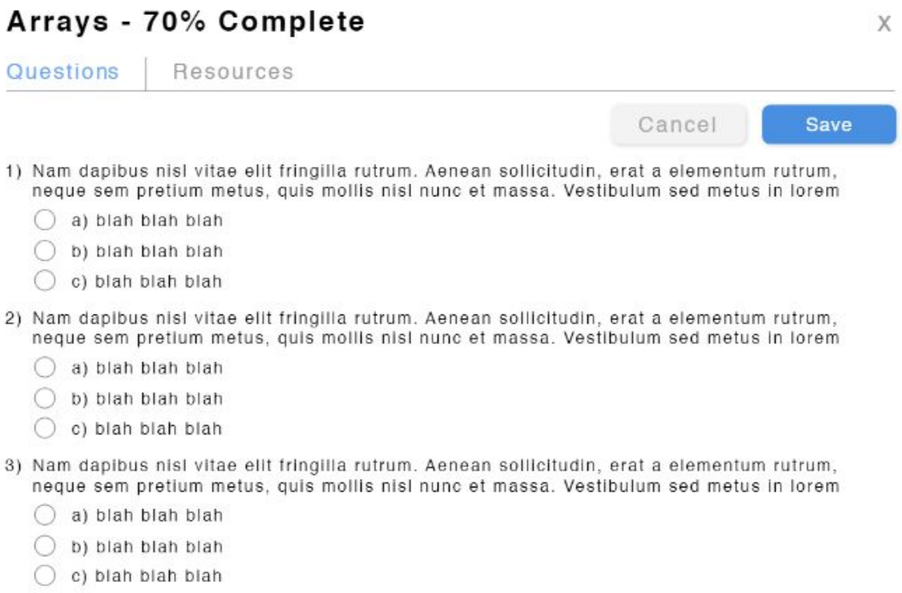


Hello

Sign in with google below

Fall ‘19 and Spring ‘20 Goals

- Scaling
  - Smarter Queries
  - Batching
  - Less Frontend Computing
- Quizzes
  - Stream Question System
  - Live Feedback
  - Novel Grading Metric



Arrays - 70% Complete

1) Nam dapibus nisl vitae elit fringilla rutrum. Aenean sollicitudin, erat a elementum rutrum, neque sem pretium metus, quis mollis nisl nunc et massa. Vestibulum sed metus in lorem

○ a) blah blah blah
○ b) blah blah blah
○ c) blah blah blah

2) Nam dapibus nisl vitae elit fringilla rutrum. Aenean sollicitudin, erat a elementum rutrum, neque sem pretium metus, quis mollis nisl nunc et massa. Vestibulum sed metus in lorem

○ a) blah blah blah
○ b) blah blah blah
○ c) blah blah blah

3) Nam dapibus nisl vitae elit fringilla rutrum. Aenean sollicitudin, erat a elementum rutrum, neque sem pretium metus, quis mollis nisl nunc et massa. Vestibulum sed metus in lorem

○ a) blah blah blah
○ b) blah blah blah
○ c) blah blah blah

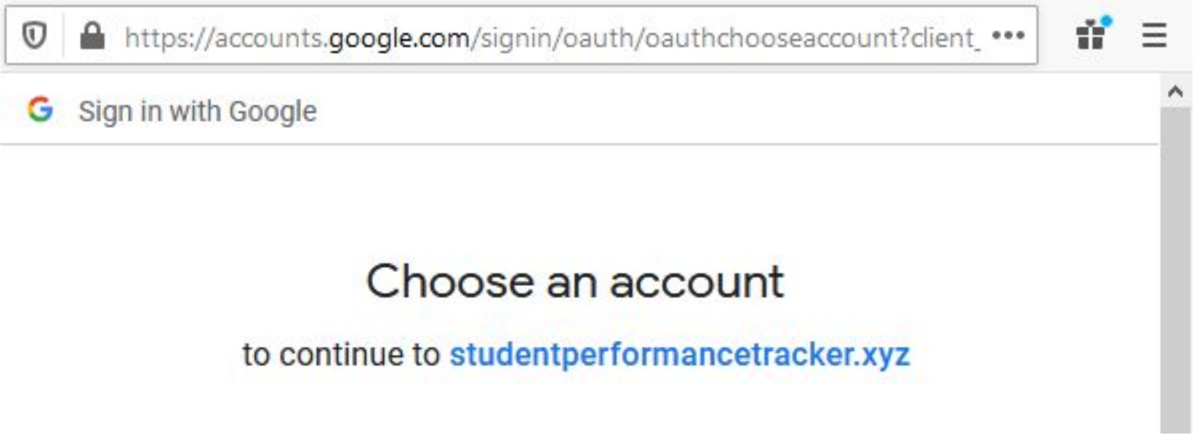
What We’re Working With

Tech & Tools Used

- Django, Django REST Framework
- Vue, Vuex
- PostgreSQL
- HTML, Javascript, CSS, Webpack
- Docker, Docker Compose
- Nginx
- Gunicorn
- Letsencrypt, Certbot
- Bash helper scripts
- Travis-CI

Backend

- Authentication Classes with OAth
- REST framework and tests all needed to authenticate properly



https://accounts.google.com/signin/oauth/oauthchooseaccount?client_

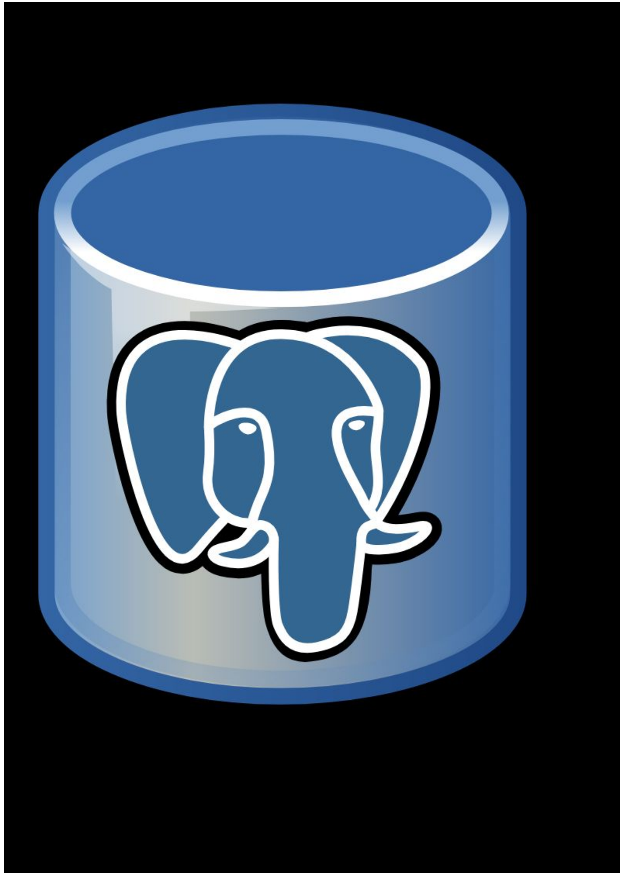
Sign in with Google

Choose an account

to continue to studentperformancetracker.xyz

Backend

- Models were re-written
- Views were re-written to work with new models
- Rewrite backend tests



Frontend Tools

- Vue/Vuex
- Javascript & HTML

Frontend Challenges

- Vuex was unfamiliar to the team at first
- Testing frontend properly was difficult
- Previous team left us with hardly any functional tests

Getting the codebase ready for production

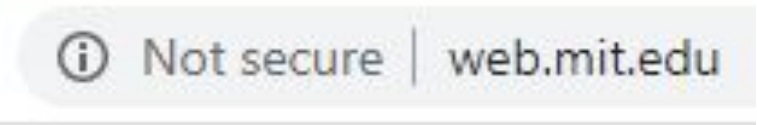
- Nginx and Gunicorn replace the django dev server
- Nginx replaces npm serve
- One bash script to enable HTTPS

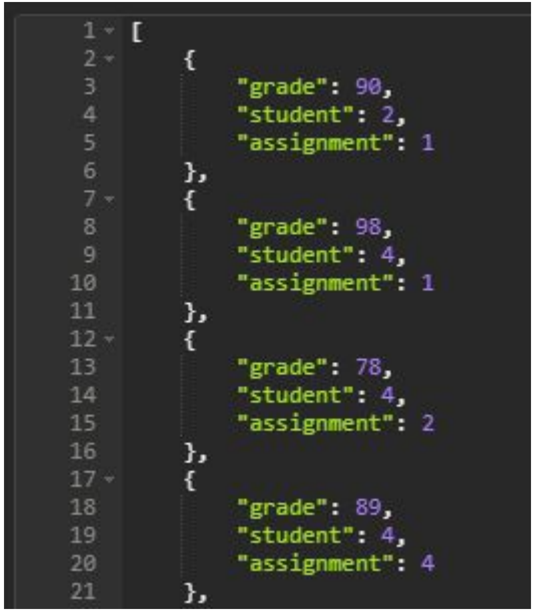
4 Goals for the Student Performance Tracker

1. Security and Authentication
2. Performance at Scale
3. Course Alignment
4. Quizzes and Grading

1. Security and Authentication

- Student grades were viewable by anyone
- Google Oauth tokens were exposed
- No HTTPS

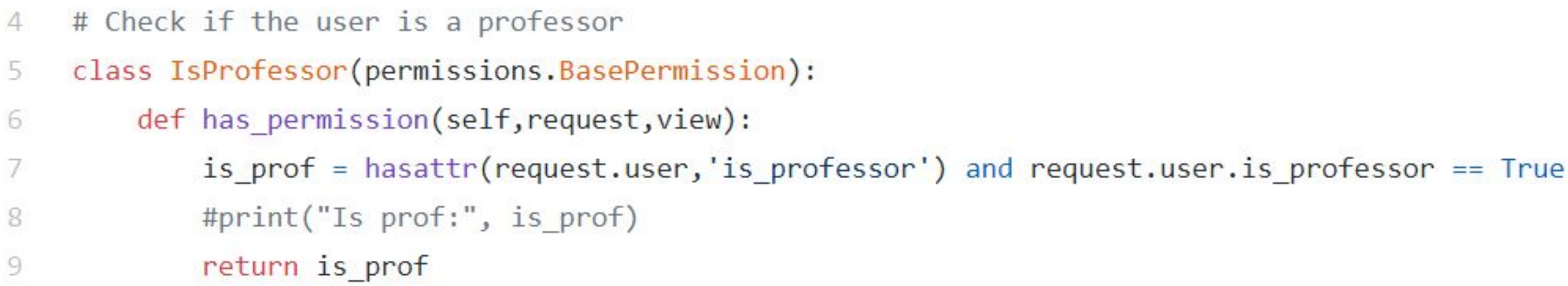




```
[
    {
        "grade": 90,
        "student": 2,
        "assignment": 1
    },
    {
        "grade": 98,
        "student": 4,
        "assignment": 1
    },
    {
        "grade": 78,
        "student": 4,
        "assignment": 2
    },
    {
        "grade": 89,
        "student": 4,
        "assignment": 4
    },
```

Django REST Framework Auth and Permissions

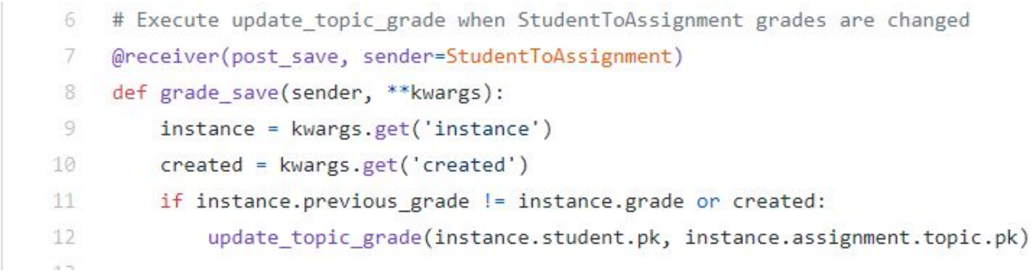
- Grades only viewable by that student, or by professors
- Removed hundreds of lines of code
- Secured passwords, tokens, quiz answers
- Added HTTPS



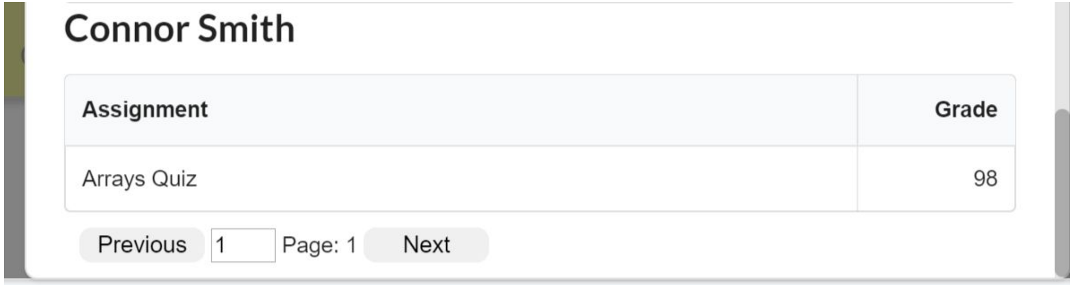
```python
# Check if the user is a professor
class IsProfessor(permissions.BasePermission):
    def has_permission(self,request,view):
        is_prof = hasattr(request.user,'is_professor') and request.user.is_professor == True
        #print("Is prof:", is_prof)
        return is_prof
```

2. Performance at Scale

- Added pagination at DB level
- Minimized data sent to frontend
- Calculate and cascade grades only when necessary
- Moved work from frontend to backend



```python
# Execute update_topic_grade when StudentToAssignment grades are changed
@receiver(post_save, sender=StudentToAssignment)
def grade_save(sender, **kwargs):
    instance = kwargs.get('instance')
    created = kwargs.get('created')
    if instance.previous_grade != instance.grade or created:
        update_topic_grade(instance.student.pk, instance.assignment.topic.pk)
```



Connor Smith

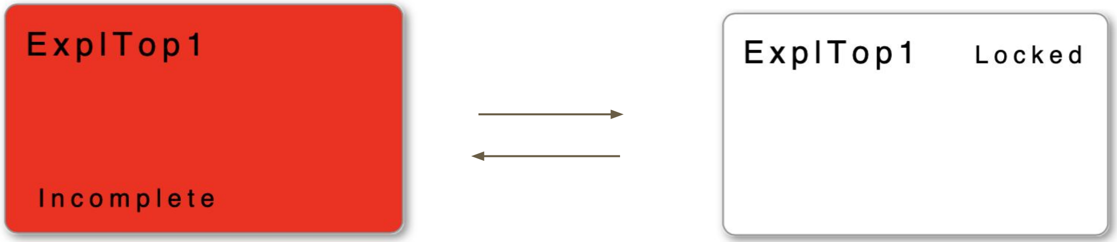
Assignment	Grade
Arrays Quiz	98

3. Course Alignment

Goal: Bring the features of the app into alignment with the professor's desired course structure.

3. Course Alignment

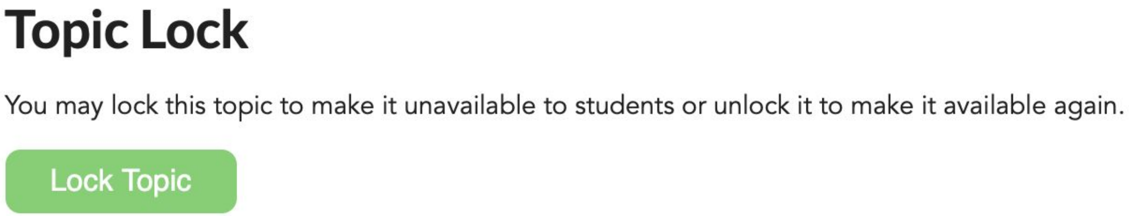
A) Topic Locking



ExplTop1

Incomplete

ExplTop1 Locked

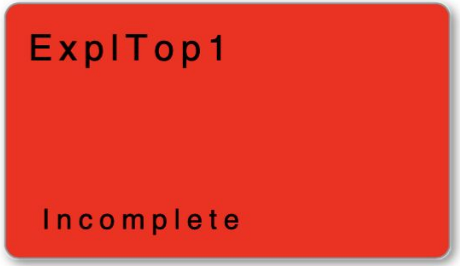


Topic Lock

You may lock this topic to make it unavailable to students or unlock it to make it available again.

3. Course Alignment

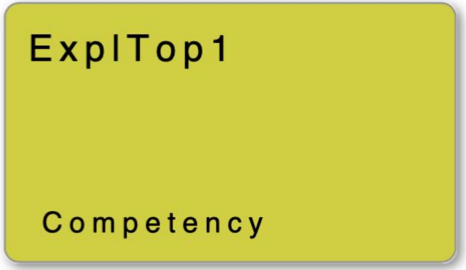
B) Grade Display and Grade Cascading



ExplTop1

Incomplete

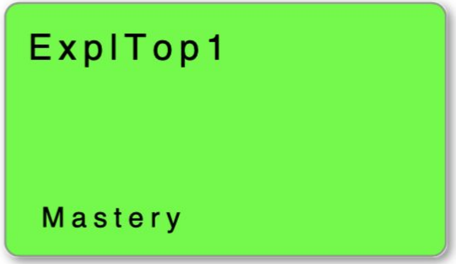
ExplTop1 - 55% Complete



ExplTop1

Competency

ExplTop1 - 77.5% Complete



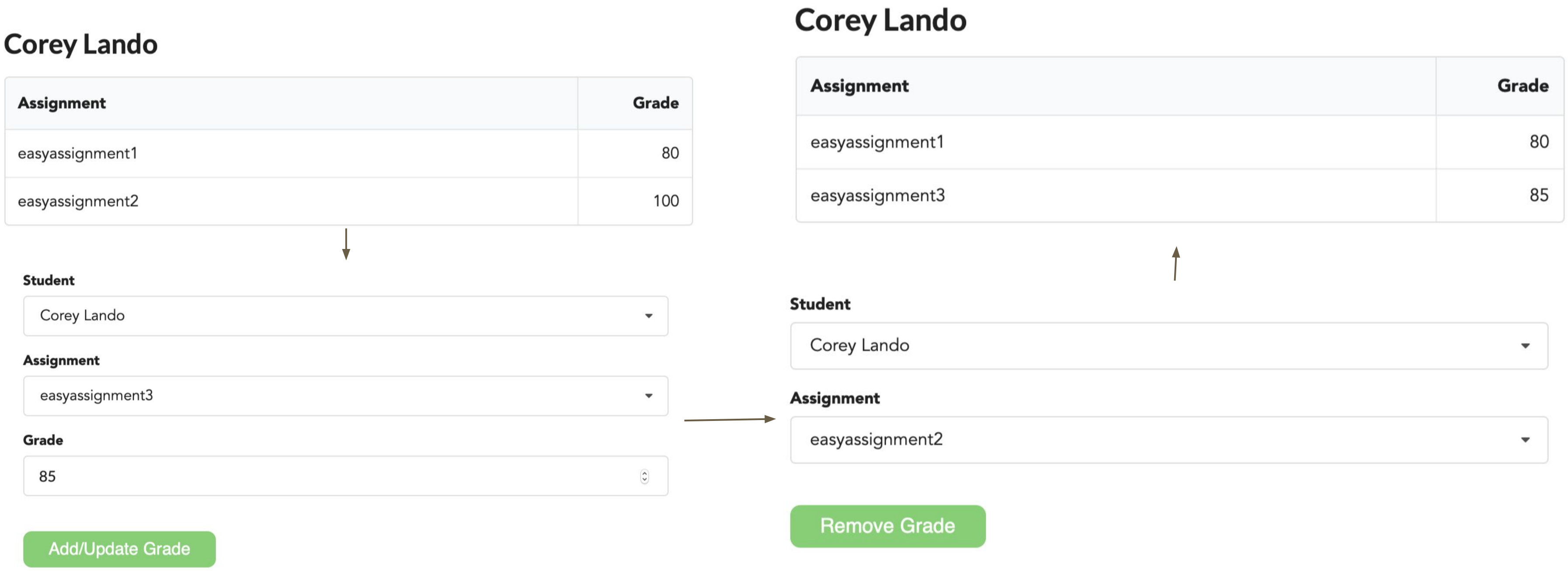
ExplTop1

Mastery

ExplTop1 - 90% Complete

3. Course Alignment

C) Tools and Individual Grading



Corey Lando

Assignment	Grade
easyassignment1	80
easyassignment2	100

Student

Corey Lando

Assignment

easyassignment3

Grade

85

Corey Lando

Assignment	Grade
easyassignment1	80
easyassignment3	85

Student

Corey Lando

Assignment

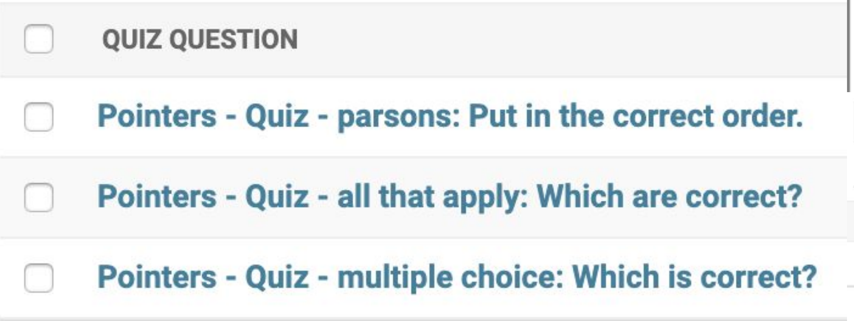
easyassignment2

3. Course Alignment

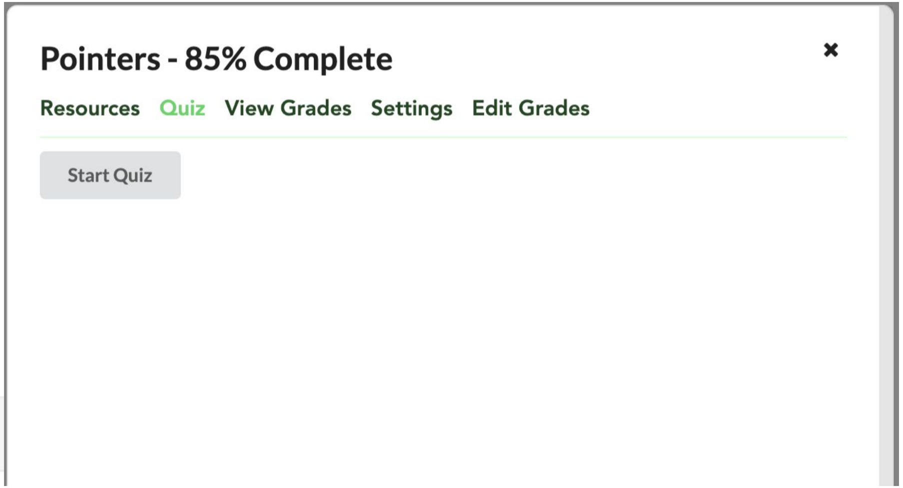
D) CSV Functionality - DEMO

4. Quizzes and Grading - Overview

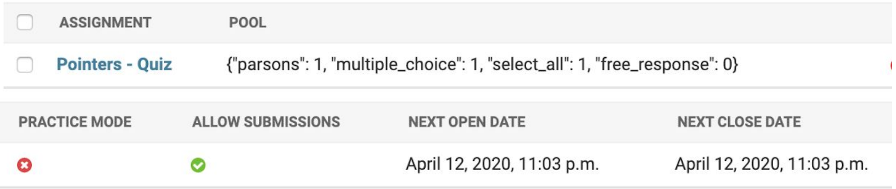
- Specialized assignment
- Pool of various types of questions
- Selects predetermined subset



□	QUIZ QUESTION
□	Pointers - Quiz - parsons: Put in the correct order.
□	Pointers - Quiz - all that apply: Which are correct?
□	Pointers - Quiz - multiple choice: Which is correct?



Pointers - 85% Complete



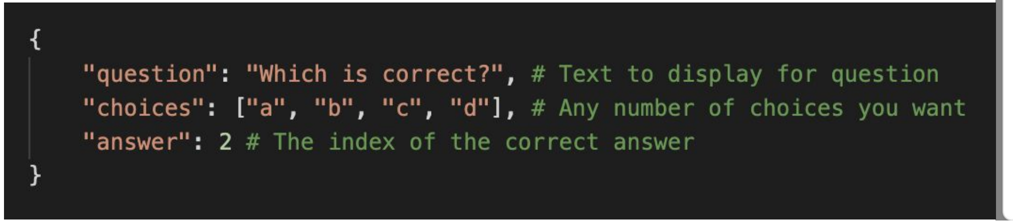
□	ASSIGNMENT	POOL
□	Pointers - Quiz	{"parsons": 1, "multiple_choice": 1, "select_all": 1, "free_response": 0}

PRACTICE MODE	ALLOW SUBMISSIONS	NEXT OPEN DATE	NEXT CLOSE DATE
☒	✓	April 12, 2020, 11:03 p.m.	April 12, 2020, 11:03 p.m.

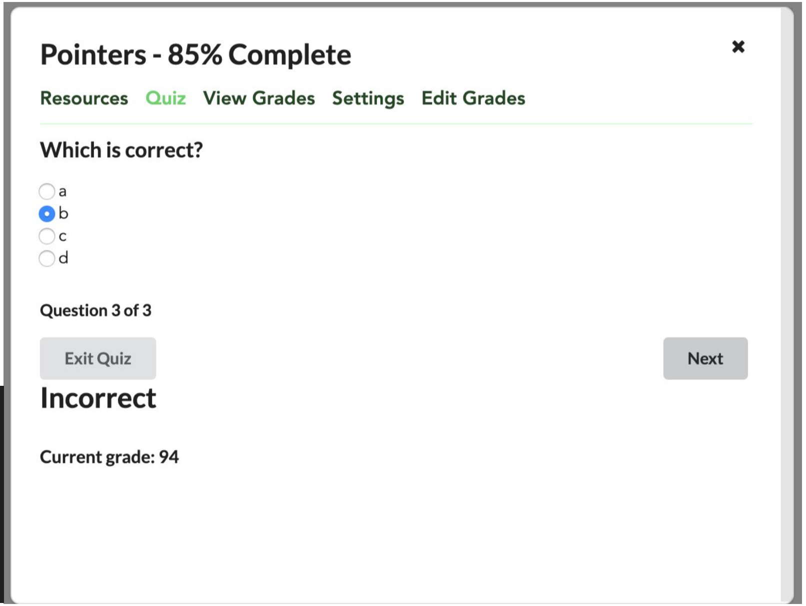
4. Quizzes and Grading - Multiple Choice

- Select 1 radio button and submit

JSON produced by CSV



```
{
    "question": "Which is correct?", # Text to display for question
    "choices": ["a", "b", "c", "d"], # Any number of choices you want
    "answer": 2 # The index of the correct answer
}
```



Pointers - 85% Complete

Which is correct?

○ a
● b
○ c
○ d

Question 3 of 3

Incorrect

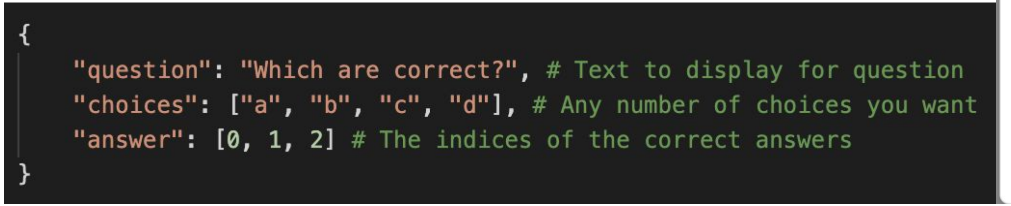
Current grade: 94

4. Quizzes and Grading - Select All That Apply

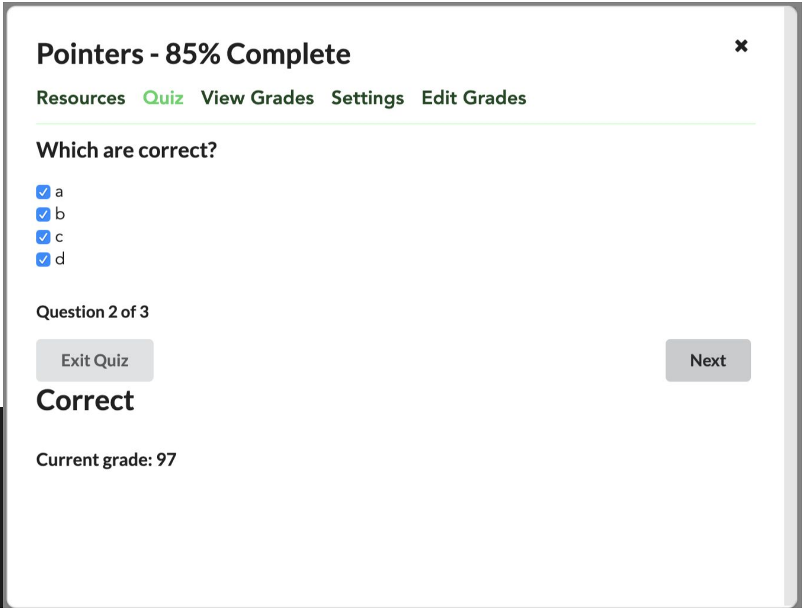
- Select any checkboxes and submit

- All or nothing (no partial credit)

JSON produced by CSV



```
{
    "question": "Which are correct?", # Text to display for question
    "choices": ["a", "b", "c", "d"], # Any number of choices you want
    "answer": [0, 1, 2] # The indices of the correct answers
}
```



Pointers - 85% Complete

Which are correct?

☑ a
☑ b
☑ c
☑ d

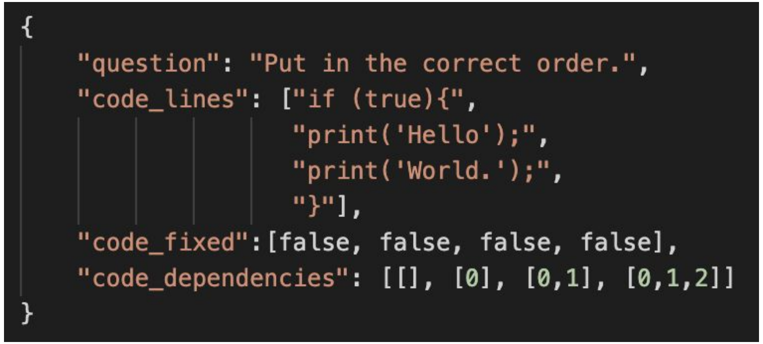
Question 2 of 3

Correct

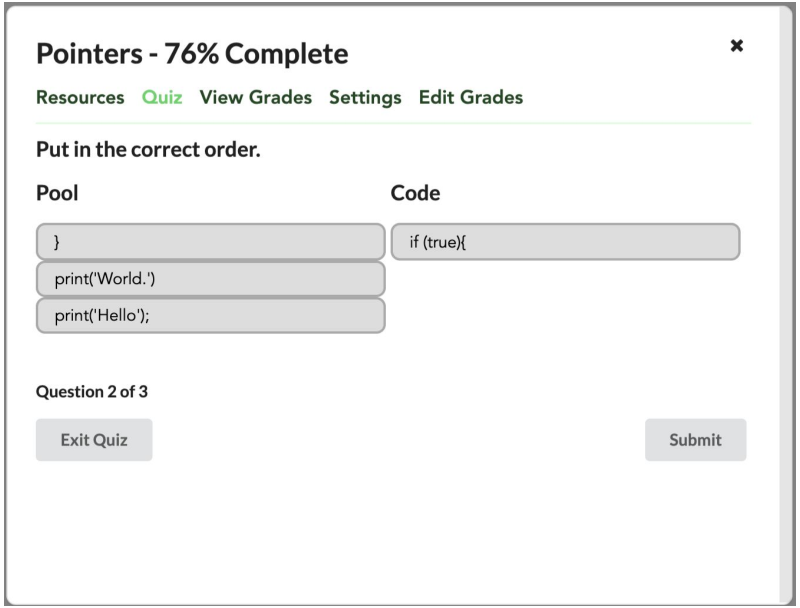
Current grade: 97

4. Quizzes and Grading - Parsons

- Click and drag code snippets
- Graded based on list of predecessors
- No “red herring” boxes



```
{
    "question": "Put in the correct order.",
    "code_lines": ["if (true){",
                   "print('Hello');",
                   "print('World.');",
                   "}"],
    "code_fixed":[false, false, false, false],
    "code_dependencies": [[], [0], [0,1], [0,1,2]]
}
```



Pointers - 76% Complete

Put in the correct order.

Pool

- }
- print('World.')
- print('Hello');

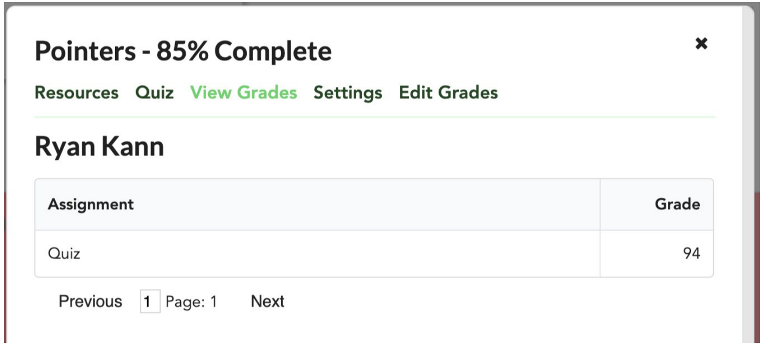
Code

- if (true){

Question 2 of 3

4. Quizzes and Grading - Grading Scheme

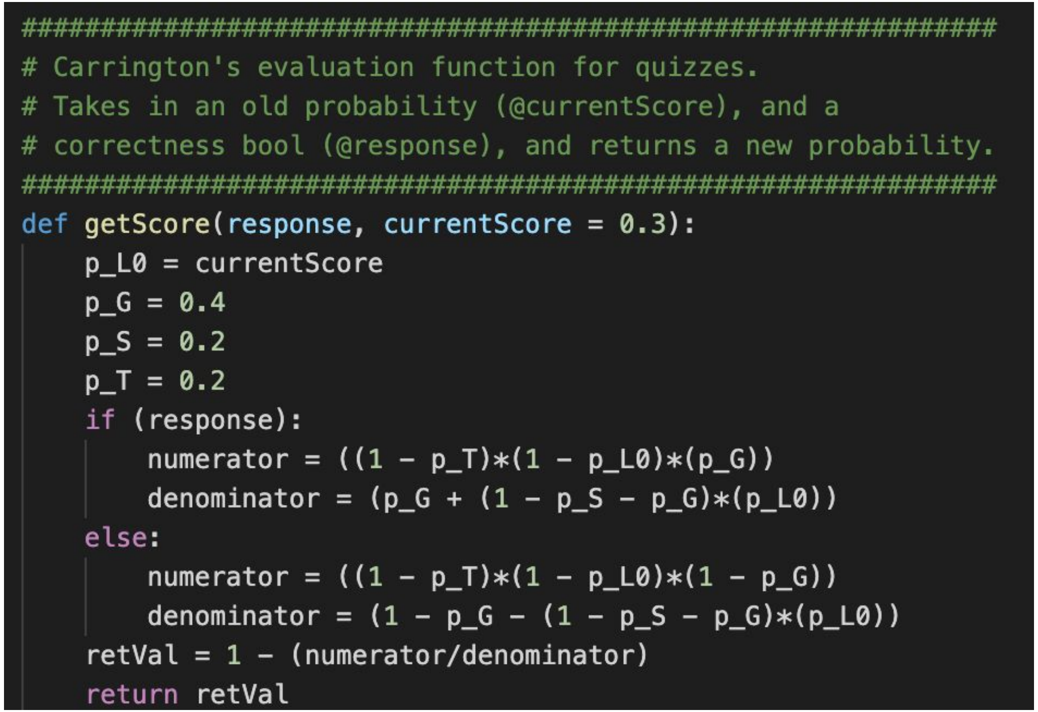
Dynamic grading based on correctness percentage over period of time.



Pointers - 85% Complete

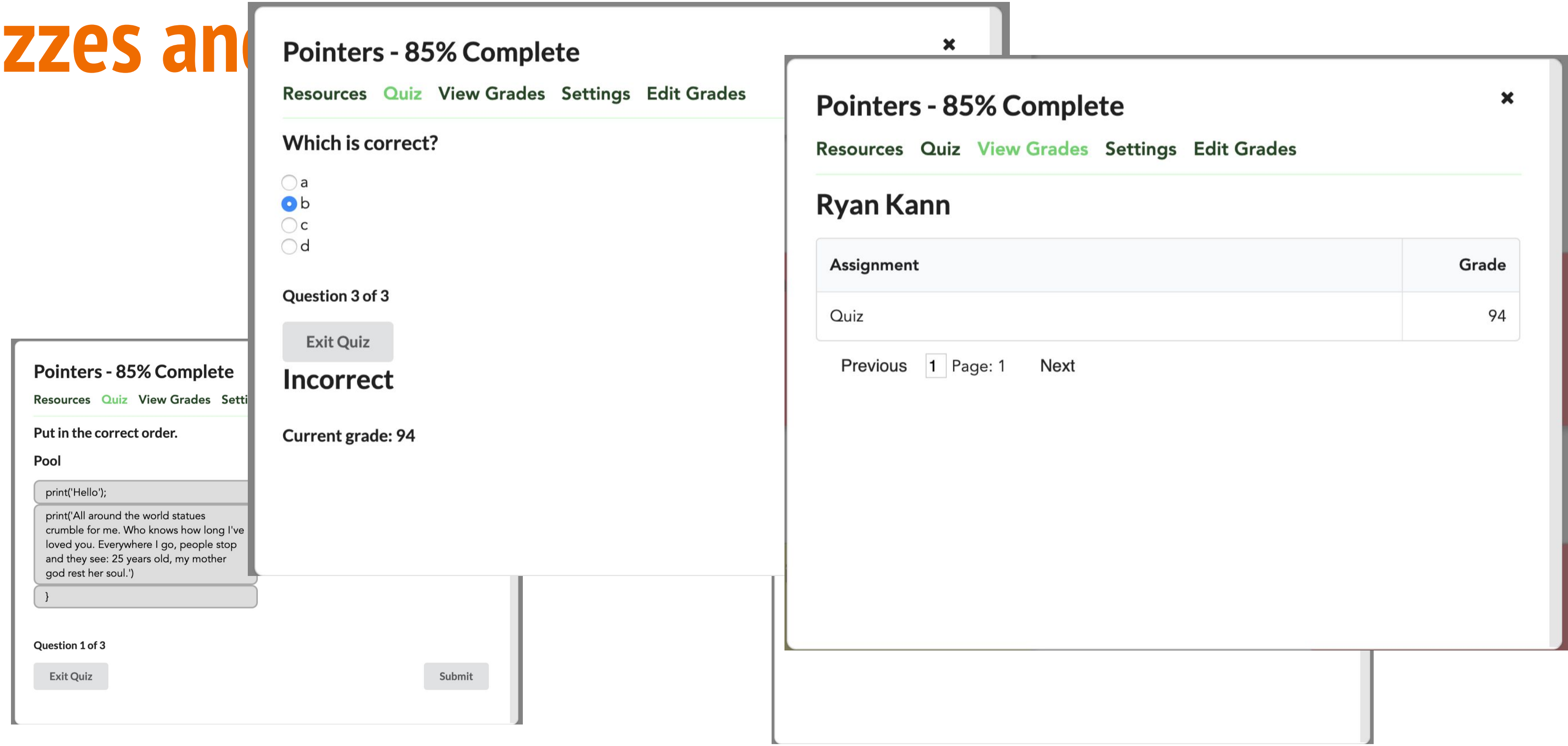
Ryan Kann

Assignment	Grade
Quiz	94



```
##############################################################
# Carrington's evaluation function for quizzes.
# Takes in an old probability (@currentScore), and a
# correctness bool (@response), and returns a new probability.
##############################################################
def getScore(response, currentScore = 0.3):
    p_L0 = currentScore
    p_G = 0.4
    p_S = 0.2
    p_T = 0.2
    if (response):
        numerator = ((1 - p_T)*(1 - p_L0)*(p_G))
        denominator = (p_G + (1 - p_S - p_G)*(p_L0))
    else:
        numerator = ((1 - p_T)*(1 - p_L0)*(1 - p_G))
        denominator = (1 - p_G - (1 - p_S - p_G)*(p_L0))
    retVal = 1 - (numerator/denominator)
    return retVal
```

4. Quizzes an



Pointers - 85% Complete

Put in the correct order.

Pool

```
print('Hello');
print('All around the world statues crumble for me. Who knows how long I've loved you. Everywhere I go, people stop and they see: 25 years old, my mother god rest her soul.')
}
```

Question 1 of 3

Pointers - 85% Complete

Which is correct?

○ a
● b
○ c
○ d

Question 3 of 3

Incorrect

Current grade: 94

Pointers - 85% Complete

Ryan Kann

Assignment	Grade
Quiz	94

1 Page: 1