S.C.O.R.E

Milestone 2

FLORIDA TECH

FLORIDA'S STEM UNIVERSITY

Team

- Charlie Collins
- Tommy Gingerelli
- Logan Klaproth
- Michael Komar

Faculty Advisor/Client

- Dr. Mohan

Milestone 2

- Implement the shell application
- Implement assignment creation
- Implement assignment submission

- Not apart of the plan
  - Implement Assignment Deletion
  - Implement Assignment View
  - Plan auto testing

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Milestone 2 - Completion Matrix

Task	Completion	Charlie	Logan	Michael	Tommy	To Do
Implement Shell Application	50%	20%	15%	50%	15%	Client-Server integration (next milestone)
Implement Assignment Creation	80%	15%	20%	15%	50%	Fix directory names and import statements on recent branch
Implement Assignment Submission	80%	40%	20%	20%	20%	Use SFTP to transfer file from client to server
Implement Assignment View	95%	20%	20%	20%	10%	Generalize to “View” both assignments and classes using flags to differentiate
Implement Assignment Deletion	100%	20%	20%	20%	40%	N/A

Implement Shell Application

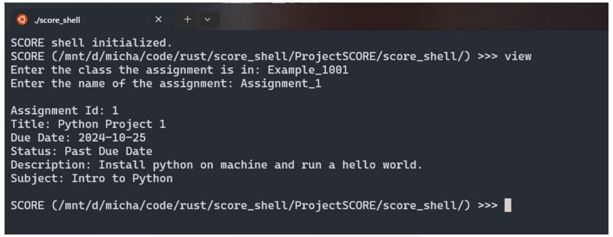
Shell Application - Completed

- The current shell mimics that of a basic terminal, with command passthrough to bash for unknown keywords.
- For custom (known) keywords, the shell runs a python script associated with the command that matched.
    - Assignment Creation, Deletion, Submit and View all work through the shell by calling their respective commands:
    - create, delete, submit, view

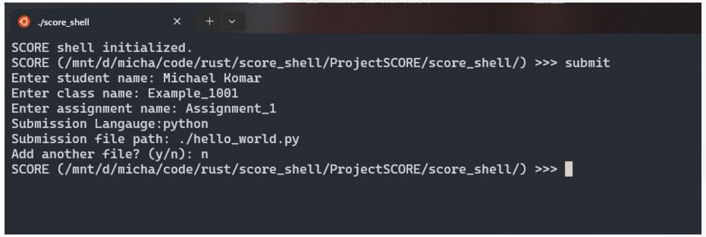
Shell Application - To-Do

- Client-Server integration
- The goal is to have the custom shell commands run on a separate machine from the one the user is on.
  - This is to keep our implementation of the modules secure
  - Only feedback will be returned from the server for a valid submission
- “Help” command to list custom commands and features of the shell.
- Possible minor quality of life changes/additions:
  - TAB Autocomplete
  - Multiple commands separated by “;”
    - This will also fix the current issue of commands being delimited by spaces.

Shell Application - Screenshots

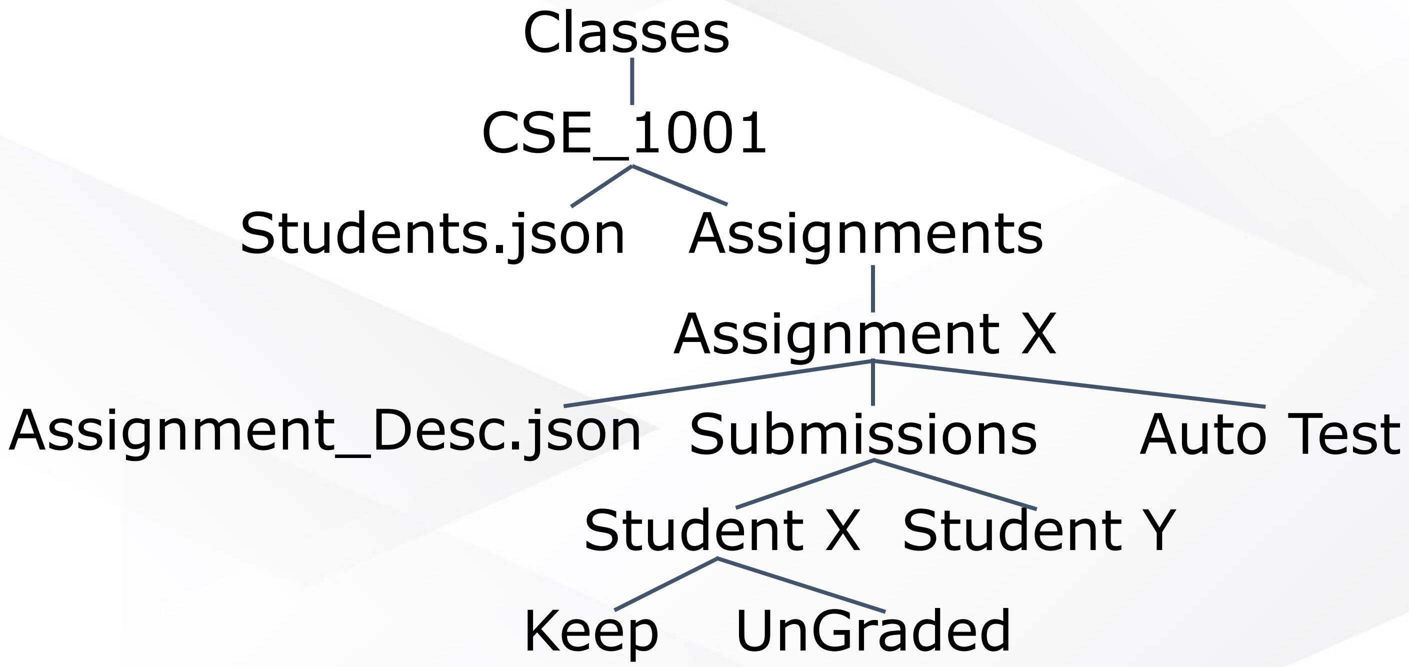


```
SCORE shell initialized.
SCORE (/mnt/d/micha/code/rust/score_shell/ProjectSCORE/score_shell/) >>> view
Enter the class the assignment is in: Example_1001
Enter the name of the assignment: Assignment_1

Assignment Id: 1
Title: Python Project 1
Due Date: 2024-10-25
Status: Past Due Date
Description: Install python on machine and run a hello world.
Subject: Intro to Python

SCORE (/mnt/d/micha/code/rust/score_shell/ProjectSCORE/score_shell/) >>>
```



```
SCORE shell initialized.
SCORE (/mnt/d/micha/code/rust/score_shell/ProjectSCORE/score_shell/) >>> submit
Enter student name: Michael Komar
Enter class name: Example_1001
Enter assignment name: Assignment_1
Submission Langauge:python
Submission file path: ./hello_world.py
Add another file? (y/n): n
SCORE (/mnt/d/micha/code/rust/score_shell/ProjectSCORE/score_shell/) >>>
```

File Structure



Implement Assignment Creation

Assignment Creation - Completed

- Takes non optional arguments such as:
  - Class, Assignment Name, Assignment Description, Due Date, Number of Attempts.
- Takes optional arguments such as:
  - useMenu, doAutoTest, checkSimililarity, acceptLate
- Arguments are currently either entirely given through command line arguments or user prompted menu.
- Once given valid input, the module creates all of the necessary directories attached to root Course directory.
  - Includes: Assignment and Submission directory, as well as a directory for each student

Assignment Creation - To-Do

- Immediate prompt when ran with no arguments.
- Accept files as an assignment description
  - PDF
  - Markdown
- Configuration of auto test and test cases

Implement Assignment Submission

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Assignment Submission - Completed

- Takes the following arguments:
  - User, Class, Assignment Name, Language, Any Number of Submission Files
- Verifies that the student is enrolled in the class and that the assignment exists
- Adds the submission to the un-graded directory
  - Wait to be auto tested
- Creates a submission description JSON
  - Submission timestamp, language, number of files in the submission
  - Will eventually have the score of the submission

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Assignment Submission - To-Do

- Implement SFTP
  - Transfer the files upon submission
- Track the number of attempts
  - Reject anything more than the limit

FLORIDA'S STEM UNIVERSITY

Implement Assignment View

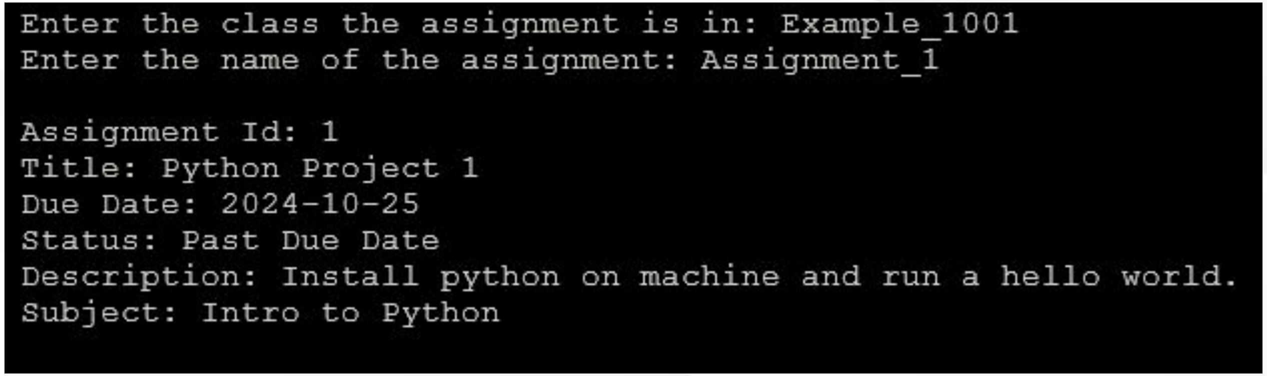
Assignment View - Completed

- Takes course and assignment name to build file path
  - Assumes description file name
- Verifies description file exists
- Parses JSON file and prints details of assignment
  - Sanitizes file input for readability

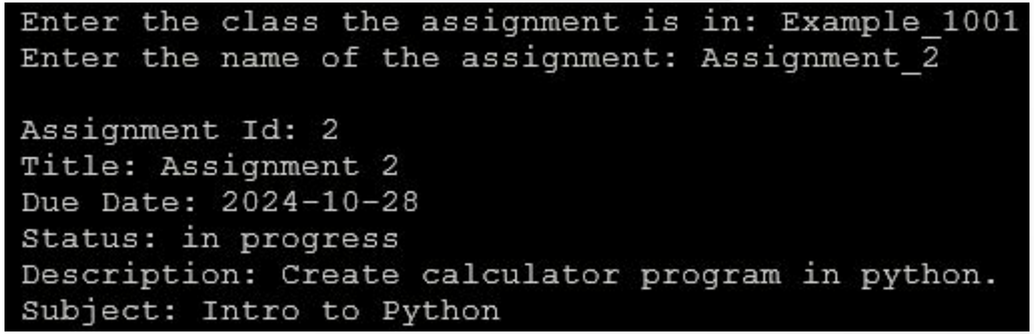
Assignment View - To-Do

- Add way to view course details
  - Considering flags as way to differentiate courses from assignments
- Finalize structure of JSON files and course folders
  - Modify view method according to JSON structure and parameters

Assignment View - Demo



```
Enter the class the assignment is in: Example_1001
Enter the name of the assignment: Assignment_1

Assignment Id: 1
Title: Python Project 1
Due Date: 2024-10-25
Status: Past Due Date
Description: Install python on machine and run a hello world.
Subject: Intro to Python
```



```
Enter the class the assignment is in: Example_1001
Enter the name of the assignment: Assignment_2

Assignment Id: 2
Title: Assignment 2
Due Date: 2024-10-28
Status: in progress
Description: Create calculator program in python.
Subject: Intro to Python
```

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Milestone 3

Task	Charlie	Logan	Michael	Tommy
Client-Server Implementation	15%	20%	50%	15%
File Transfer	50%	10%	30%	10%
Auto Testing	20%	20%	10%	50%
Feedback System	20%	50%	10%	20%