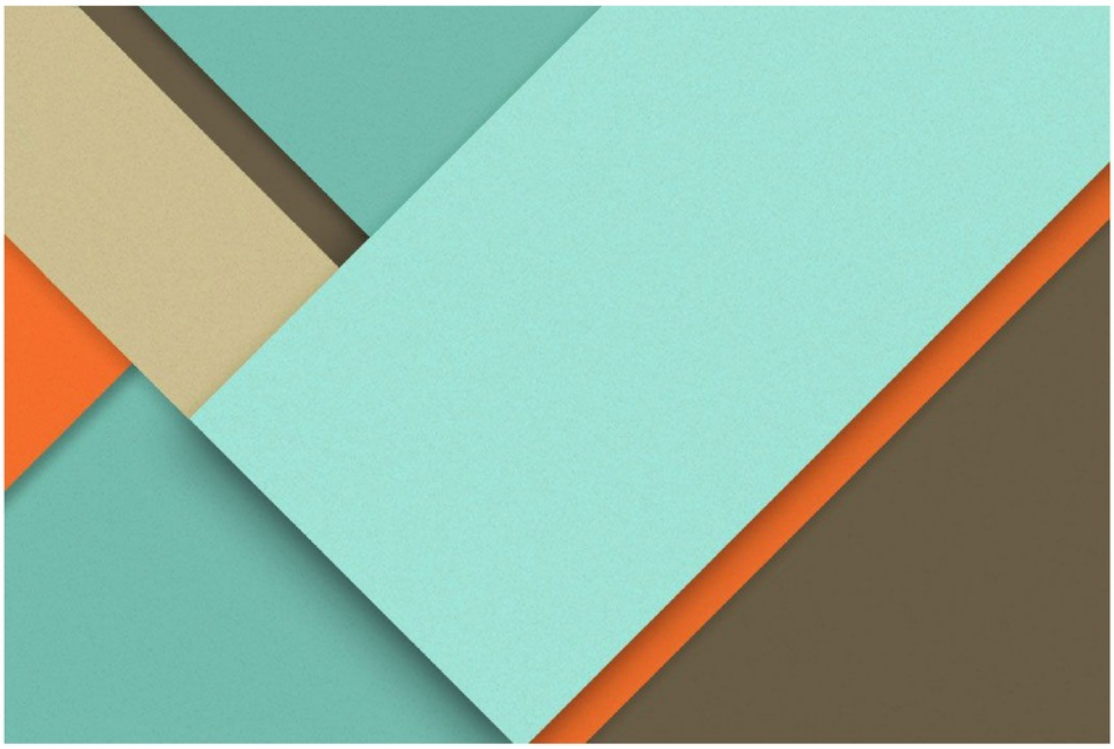
Cloud Computing - Project Report

by Devante Wilson

Submission date: 09-Apr-2019 02:07PM (UTC-0400)
Submission ID: 1109103117
File name: Cloud_Computing_-_Project_Report.pdf (723.77K)
Word count: 2177
Character count: 10786



StudyBuddy

Sikandar Shahbaz | Ebrahim Merchant | Emin Avenessian-Zadeh | Devante Wilson
Group 1
Cloud Computing - Term Project Report

Summary

For our project, we were motivated to create something that would help students such as ourselves, who much too often feel as if we are lost, and without help, when studying. The issue arises when we confront a topic that we do not quite understand, and the topic also has very few resources online to help us understand it. In that case, there would be nothing more helpful than a platform where we could ask for help and maybe even get a quick peek at someone else's notes. So our group decided why not create such an online service that achieves exactly that. We created a web application that is focused on building a virtual space for students to collaborate, and help each other out by sharing notes and answering questions for one another. We focused on ensuring that users have a very easy time navigating through the website, and do not feel hesitant to ask, or answer questions. It is meant to be a virtual space that students can see as their very own study companion.

Introduction

We created a web application that allows students to discuss topics related to their education field, as well as share useful notes to help each other during study periods. This application is designed to assist students in their program, and to help alleviate the burden of group studying online. Not only can a student collaborate with the students at their own university but, they are also able to work with students from other schools. The target audience is any student at a post-secondary educational institution (University/College) but, more specifically the students that feel as if they need help in achieving their educational goals. Observing the following hypothetical use case, if a student feels that they need help understanding "Dijkstra's Algorithm" for an algorithm design course, they can simply log into their account, and join a public virtual room to ask for help with the subject.

The application will have two main options set for the user upon logging-in. The first one being 'Join a public room'. This option is completely free-of-charge for the user and allows the user to enter themselves into an already existing room that revolves around a certain topic. A registered user can do this by searching the name of a room (ex. 'Algorithm Design Help') to a corresponding chat room. The other option presented to the user is the 'Create a private room' option. This exists so that students who need to carry-out a more intensive research/study on a topic can a have a private virtual space where they can really dig deep into a specific topic amongst trusted, and experienced students rather than a public room where anyone can come, and go as they wish. The drawback to this that if someone wants to create a private room, they must pay a small fee in order to do so.

We also hope that this product will help against students procrastinating on their work. Often times, the main reason students procrastinate is because they have a certain lack of

resources for help. With the power of this application, students will have 24/7 access to study assistance that will allow them to find the helpful resources they need in order to get their work done on time. The more someone comments on other people's questions, the higher priority they will be given, and when they themselves need help, their questions will be put atop of the questions board. This is an efficient way of convincing users to not only seek help but, offer it as well.

Background and Related Work

There are some related works to this application but, there is no major online platform that provides the exact same service that we provide. For instance, the world famous web-forum known as *Reddit* has many sub-categories that allow people to seek study help from each other, and share notes. However, this is very different from our product as it is solely focused on university students collaborating for the sake of helping each other, rather than a web-forum that allows people to create sub-categories. Furthermore, we provide students with private virtual rooms where they can message and share notes amongst each other securely.

Another related work that already exists is *StackOverflow*; it is a web-forum for software developers to provide help to fellow developers, and to solve any potential problems they may be having with their code. Our product is not focused towards developers/engineering students but, rather students of all fields. Students studying any major can utilize our product to study and communicate for their course work.

There are also other very similar services such as *StudyDrive*, *StuDocu,* and *OneClass*. In order to use *StudyDrive*, a user would need to log in, and select the university they attend. After that, they will be able to find documents, and notes that pertain to their courses. They currently have no live chatting service. *StuDocu* provides the same service but, it also has a question & answer service that is not live. It also provides some free study guides to help students understand how to study more effectively. *OneClass* allows students to share notes and other resources, and also allows students to ask questions from experts that have extensive knowledge in that field. All of the platforms stated above have one very glaring missing piece that our application does have and that is the ability to not only share resources with students from one's own university but, to also share resources with those from other universities. This specific feature is very unique and special for a lot of reasons. Not only does it open up the horizon of someone's educational resources but, it also allows students to look at topics from much different perspectives. Students studying the same course at the same university will approach a problem the exact same way due to their similar educational conditioning. However, students studying the same course from separate universities will definitely see the problem from different views. So in some ways,

working with students from other educational institutions is far more beneficial than studying with your own colleagues.

Proposed Solution

We feel that the best solution to the problem of students feeling overwhelmed is group study, and collaboration with other students. This is so the struggling student does not feel that they are going through this rigorous educational journey alone. Unfortunately, it is not always possible for people to physically meet up with each other due to proximity and transportation difficulties. Luckily, this issue is avoidable through the power of cloud computing.

We have created a virtual environment where students can receive help from their peers for specific topics, or courses where they feel they need help. Users will have the option to chat with each other on a public page that is viewable by anyone or have a more closed study session using the private room service. Another feature that will be available is the sharing of personal notes. Students will be able to compare, and contrast their notes; this will allow them to better understand the topic, and get a stronger grasp of the concepts. Perhaps the most useful feature that is available through this application is the quiz feature. Students will able to test each other using this feature to better prepare for a real-life examination. The quiz feature has multiple formats of implementation; True/False, Multiple choice, Short Answers, as well as Fill in the blank. These are all of the same examination procedures from real world courses. In a more specific scenario, one student will be able to post a question and have the answer invisible to the other user(s), and the student being tested will select or write down an answer. If the two answers match, the testee will be told they are correct and if not, they will be told the correct answer. If the testee desires, they can ask the tester for an explanation of how they arrived to the correct answer.

This application was designed in a very calculated and organized manner. Starting with the HTML code in the front-end to construct the layout. This was quite simple as the tables/design needed to visualize the application is not complicated at all. It is consists of simple HTML objects that allow the user to sign-in/log-in, and either create a room, or join a room. Next, we used a CSS library called Materialize that allows the programmer to code a certain visual design only once, and then they will be able to call it from other areas in the application if needed, rather than writing the code again. This helped us make the application source code more efficient, and even reduce the web-page load time due to less amount of resulting code.

The backend was done using Python. There is a specific framework for web communication between clients called Sockets IO. We implemented this using Python because new-school

frameworks, and libraries are easily implemented in Python and there is also quite a bit of online documentation that helped with the task.

We used digital ocean as an Infrastructure-as-a-Service to help deploy the application to the cloud.

The final technology used in this application was Google Firebase. What is so unique about Firebase is that it can be accessed from the front-end unlike most Databases where you have to write code in the backend to access the Database. Firebase has real-time data stored as JSON objects, and everything is entirely cloud-hosted which naturally makes it the default choice for anyone developing a cloud application.

Evaluation and Results

The application will have a very simple service operation that will allow the platform to earn enough revenue to be self-sufficient. There will be a month-to-month subscription model that will enable the user to take advantage of additional features that are offered. Creating a private room is one of those features. Users will want to create a private room if they are looking for a more private space to contact other students and this service will have a small fee. There will also be a premium quiz mode that costs a small fee and it will allow users to enable some more features when assigning an taking quizzes. Another source of revenue will be the Ad Revenue that will be generated through sponsors and clients wishing to market their own products on our platform. All of these combined should add up to enough of a revenue for the application to continue functioning optimally. Another issue that needs to be addressed is the possibility of academic misconduct. At first glance, it seems that it is entirely possible for users to share sensitive information that result in cheating. This will be prevented through anti-plagiarism API that prevents two different users from submitting the exact same solutions. There are many free open-source APIs online that detect for potential fraudulent or copied work in submissions.

Test Cases

As a part of the evaluation and results phase, our team wrote a series of test cases to ensure the functionality of the implemented features of the web application. We utilized the *Selenium* testing framework for web applications; it allowed us to write functional test cases and run them in a browser instance driver called a 'WebDriver'. Once executed, the test cases are fully automated with the browser instance being visible, and with it mimicking the actions of a user.

Test: login


```python
# import required libraries
from selenium import webdriver
from selenium.common.exceptions import TimeoutException
from selenium.webdriver.common.keys import Keys

# define web browser driver to use
driver = webdriver.Chrome(r'C:\Users\100554361\Documents\GitHub\cloudproject\drivers\chromedriver.exe')
# set page load timeout to seconds
driver.implicitly_wait(30)
# load specified web page in a browser session
driver.get('http://68.183.207.31:5000/')

# set the credentials to test
email = ''
password = ''

# attempt to login via homepage
try:
    # click login button upon presence of login button
    driver.execute_script("document.querySelectorAll('#login')[0].click()")
    driver.execute_script("document.querySelectorAll('#loginBtn')[0].click()")
    print ('Login page is ready.')

    # locate the email address field and enter the email
    driver.find_element_by_id("emailLogin").send_keys(email)
    print('Email entered into form.')

    # locate the password field and enter the password
    driver.find_element_by_id("passwordLogin").send_keys(password)
    print('Password entered into form.')

    # locate the sumbit button and submit the form
    driver.find_element_by_id("loginBtn").click()
    print('Logged in successfully.')

    # locate the 'join a public room' button and click it
    # driver.find_element_by_id("joinRoom").send_keys(Keys.ENTER)
    # print('Clicked the "join a public room" button.')

    # select an available room from the room list
    # driver.execute_script("document.querySelectorAll('#rooms')[0].click()")
    # print('Selected an available room.')

    # quit the driver and close every associated window
    #driver.quit()

except TimeoutException:
    print('Loading of the "login" element took too long.')
```

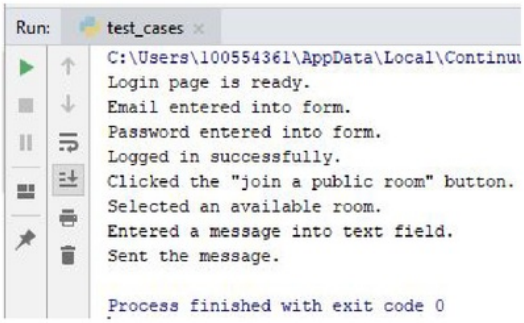
Test: join a room


```
def join_room_test():
    # locate the 'join a public room' button and click it
    driver.find_element_by_id("joinRoom").send_keys(Keys.ENTER)
    print('Clicked the "join a public room" button.')

    # select an available room from the room list
    driver.find_element_by_class_name('collection-item').click()
    print('Selected an available room.')
```

```
# attempt to execute test cases
try:
    login_test()
    join_room_test()
    #send_message_test()

    # quit the driver and close every associated window
    #driver.quit()

except TimeoutException:
    print('Loading of the "login" element took too long.')
```

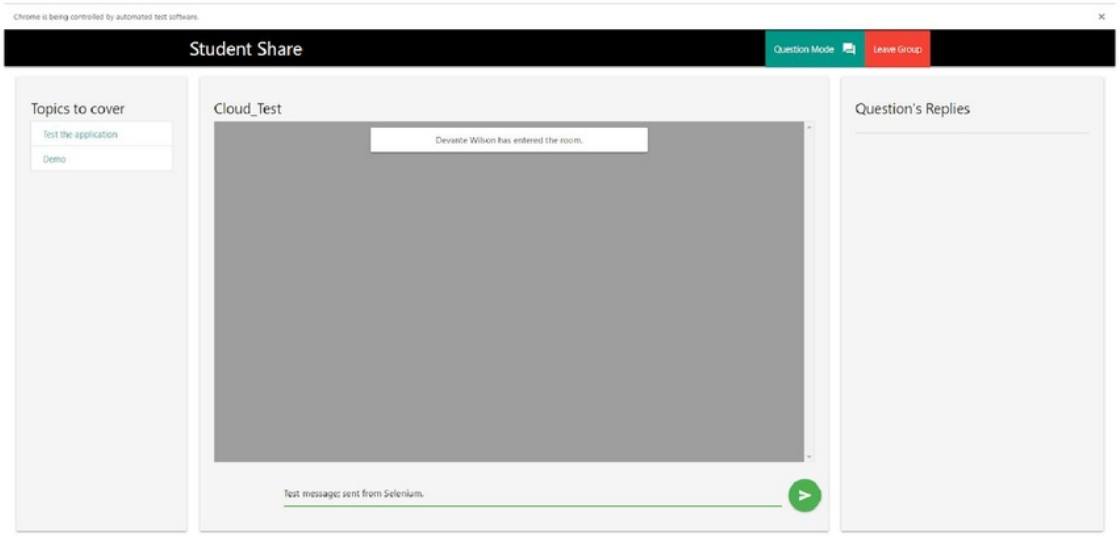
Test: send a message with button click

In this test, a fault was caught. Clicking on the 'send' button in the chat room web page had no effect. Instead, users can press 'enter' on their keyboard to send messages.




```
def send_message_test():
    # define a test message
    testMessage = 'Test message; sent from Selenium.'
    # locate the 'message' text field and type message
    driver.find_element_by_id('message').send_keys(testMessage)
    print('Entered a message into text field.')
    # locate the send button and click it
    driver.find_element_by_name('action').click()
    print('Sent the message.')
```

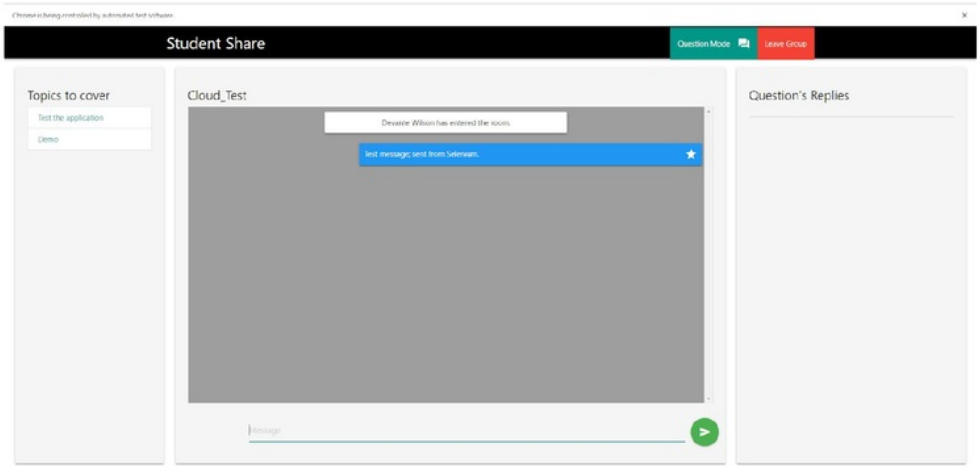
```
# attempt to execute test cases
try:
    login_test()
    join_room_test()
    send_message_test()

    # quit the driver and close every associated window
    #driver.quit()

except TimeoutException:
    print('Loading of the "login" element took too long.')
```



Test: send a message with enter key

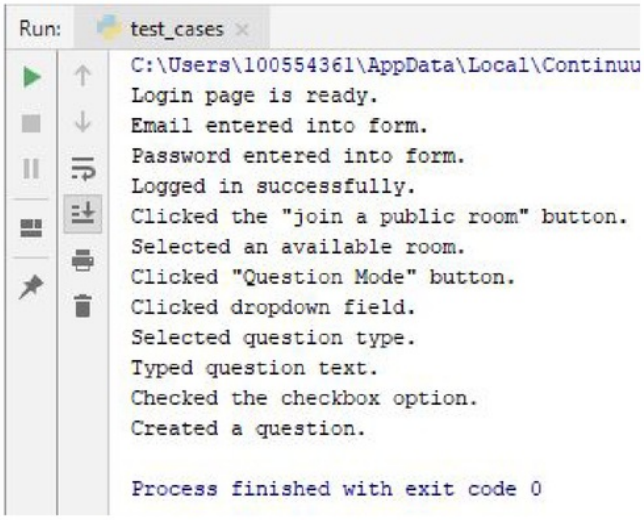
As mentioned above, the 'send' button click had no effect. Test case was modified with the enter key responsible for sending message. The expected effect was produced.

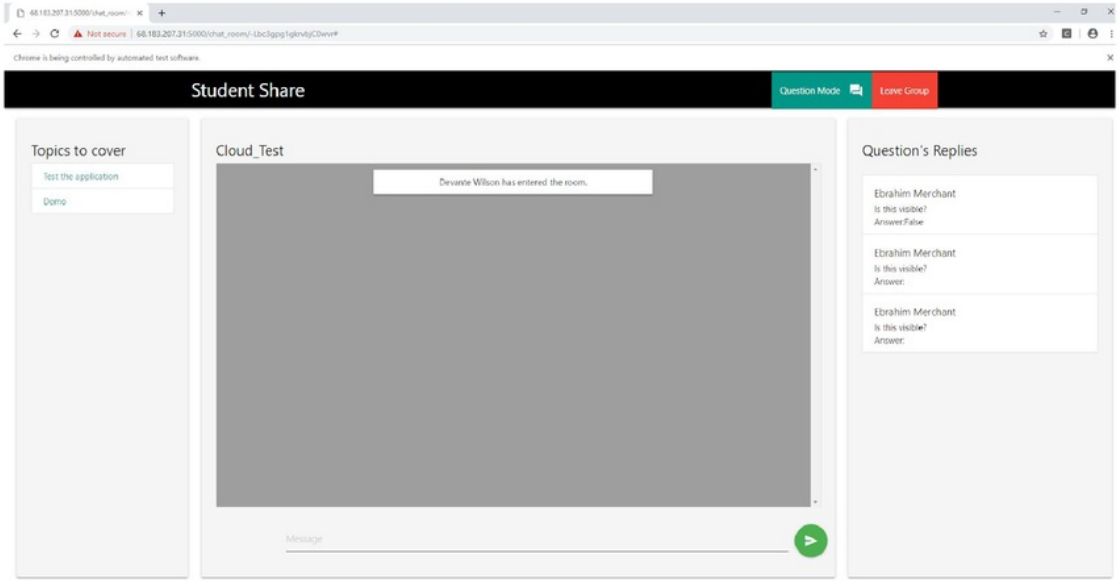
```
def send_message_test():
    # define a test message
    testMessage = 'Test message; sent from Selenium.'
    # locate the 'message' text field and type message
    driver.find_element_by_id('message').send_keys(testMessage)
    print('Entered a message into text field.')
    # locate the send button and click it
    #driver.find_element_by_name('action').click()
    driver.find_element_by_id('message').send_keys(Keys.ENTER)
    print('Sent the message.')
```



Test: create a question

```
def create_question_test():
    # locate the 'question mode' button and click it
    driver.find_element_by_id('startQuestion').click()
    print('Clicked "Question Mode" button.')
    # locate the 'choose question type' drop down box and click it
    questionType = driver.find_elements_by_class_name('select-dropdown')
    questionType[0].click()
    print('Clicked dropdown field.')
    # locate the 'True/False' option and click it
    driver.execute_script('$("#qType").val("1").trigger("change");')
    print('Selected question type.')
    # locate the 'question' text field and type question
    questionMessage = 'Is this visible?'
    driver.find_element_by_id('question').send_keys(questionMessage)
    print('Typed question text.')
    # locate the 'true' checkbox and click it
    driver.execute_script('$("#true").prop("checked", true)')
    print('Checked the checkbox option.')
    # click the 'submit' button
    driver.find_element_by_id('sendQuestions').click()
    print('Created a question.')

# attempt to execute test cases
try:
    login_test()
    join_room_test()
    #send_message_test()
    create_question_test()
```





Conclusion and Future Work

In conclusion, it is a known fact that group studying and collaboration are the best way to learn a subject. The issue is that it is not always a viable option for students for many reasons such as transportation and logistics. Our group has great reason to believe that we have created an application that can revolutionize the way modern-day students learn. It is a product that can help change the education industry forever through its unique ideas and services that are hard to find anywhere else. It is obvious that there are other similar products that students can use but, none offer the wide variety of features that this one

does. In the near future, we hope that we can upscale this application so that many users can take advantage of its unique features to enhance their own educational careers.

References

https://www.studydrive.net/

https://www.studocu.com/

https://oneclass.com/

https://materializecss.com/

Cloud Computing - Project Report

ORIGINALITY REPORT

0%	0%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		