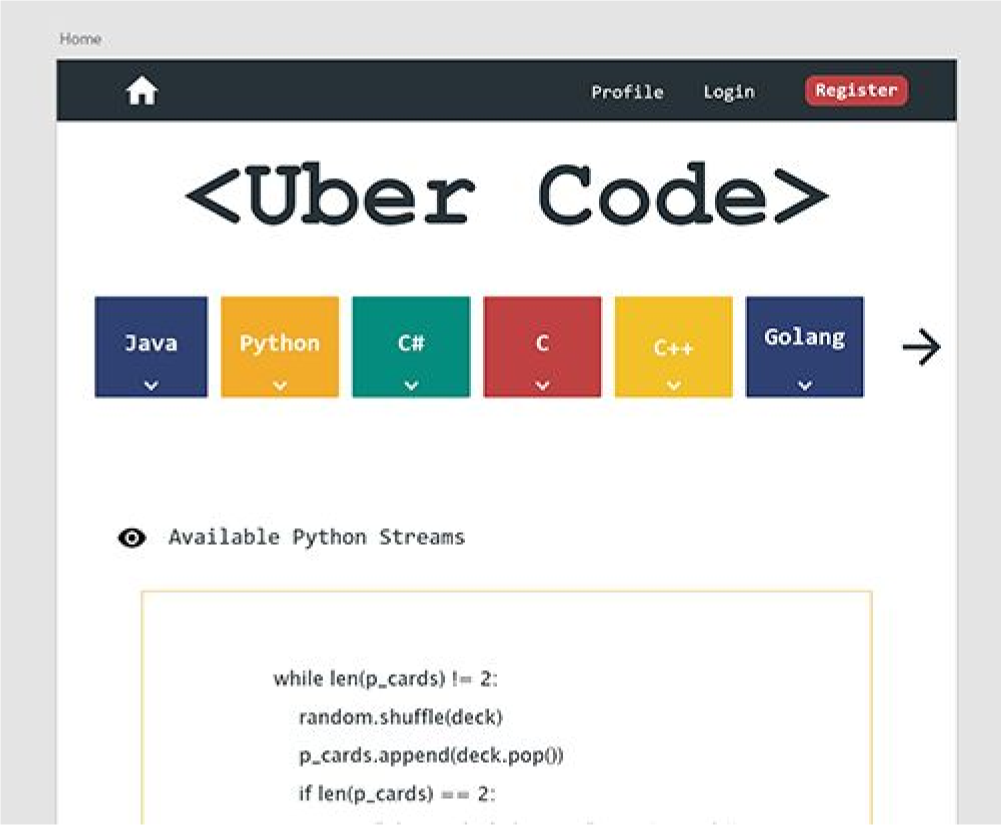
CSCI443 Final Project - Spring 2020

Addison Boyer, Colton Gerth, Mark Lannen

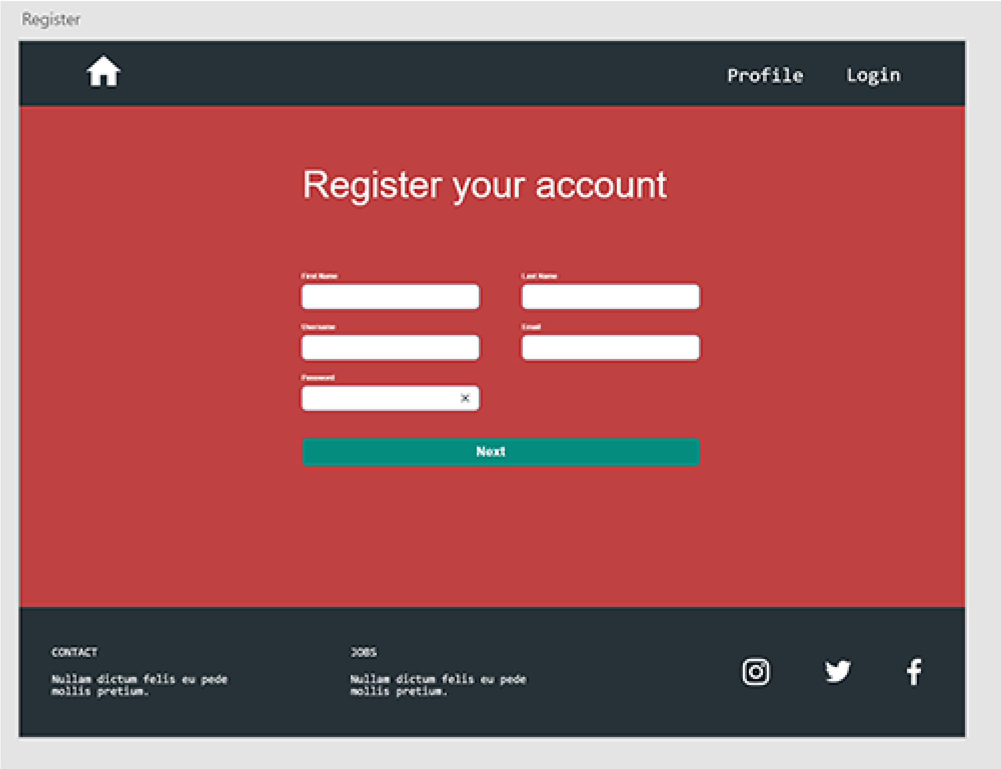
<Uber Code>

1. **Introduction -** UberCode is an online tool for watching and demonstrating coding skills. Think of it as ‘Twitch’ for coding!
    a. **Project Description -** Users can log on to Uber Code and view either a tutorial or a coding competition in their chosen language. The intended audience of UberCode is programmers at all levels of abilities. Beginning programmers can watch and learn from more experienced programmers. And experienced programmers can share their skills or test their abilities against other programmers of similar skill levels.
    b. The primary goal of product testing was to gauge the ease of use of UberCode, the user’s ability to navigate around the site - register, login, choose a coding language and answer questions about what specific coding streams or tutorials.
2. **Methodology**
    a. **Overview of goals and testing procedure -** Our main goal was to gauge the ease of use of UberCode by conducting controlled tests where users were given a series of tasks to perform using a prototype of UberCode built by Adobe xd.
    b. **Functional and Non-Functional Requirements -** Users should be able to:
        i. Go to home page



```
while len(p_cards) != 2:
    random.shuffle(deck)
    p_cards.append(deck.pop())
    if len(p_cards) == 2:
```

ii. Register.



Register

Register your account

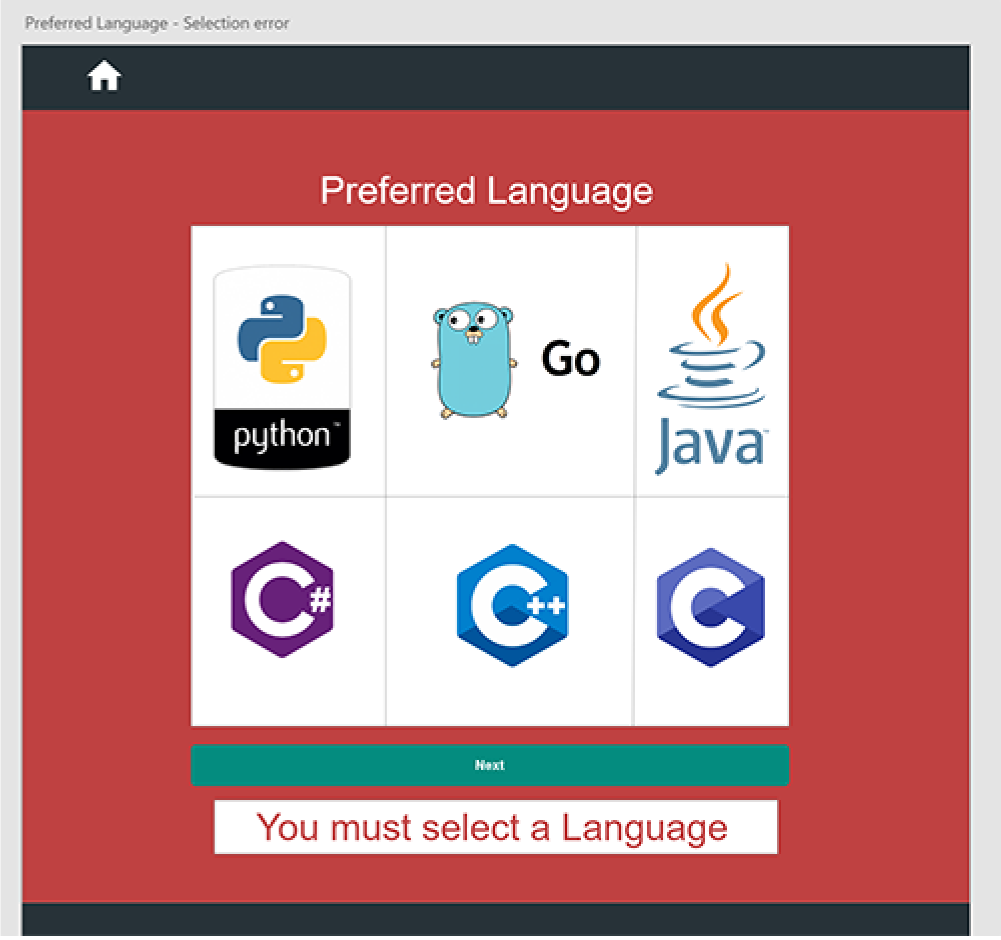
CONTACT

Nullam dictum felis eu pede mollis pretium.

JOBS

Nullam dictum felis eu pede mollis pretium.

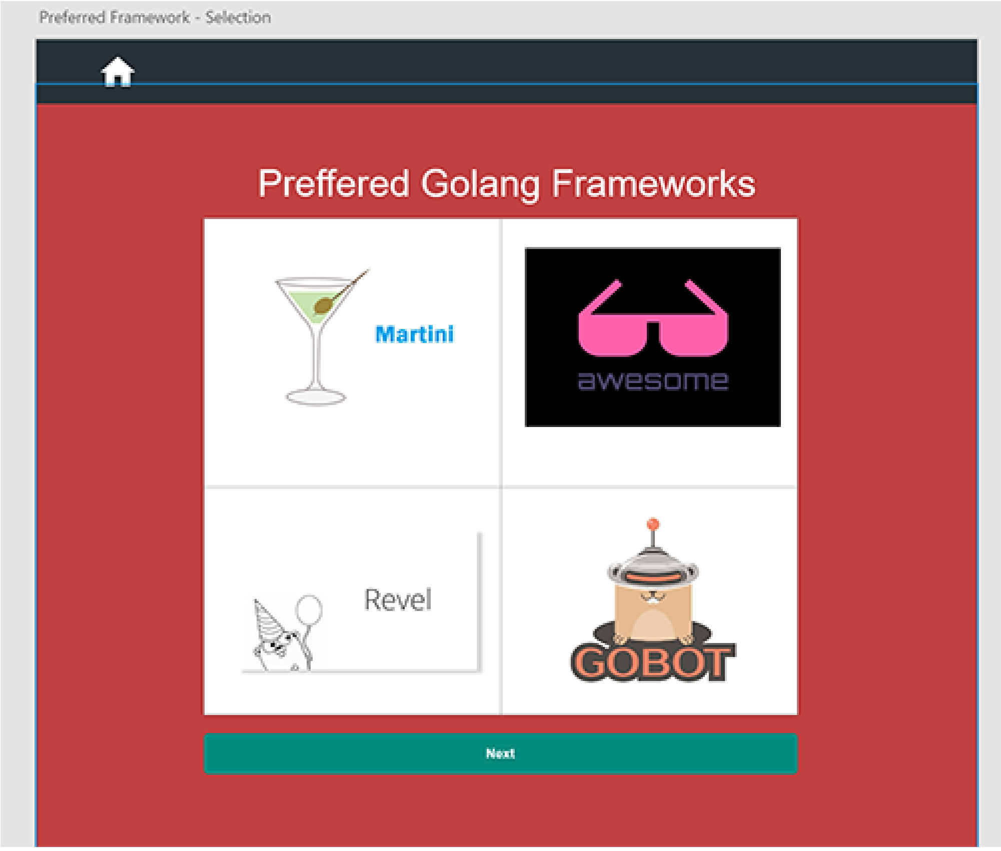
iii. Choose a preferred programming language and framework.



Preferred Language - Selection error

Preferred Language

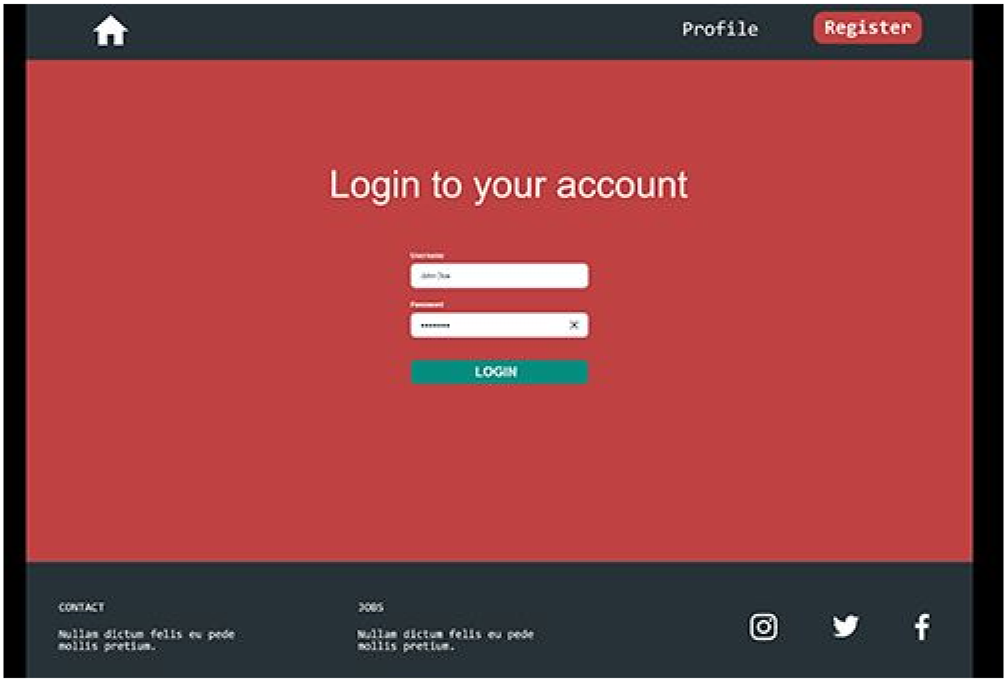
You must select a Language



Preferred Framework - Selection

Preffered Golang Frameworks

iv. Login.



Login to your account

Username: John Doe

Password: ••••••••

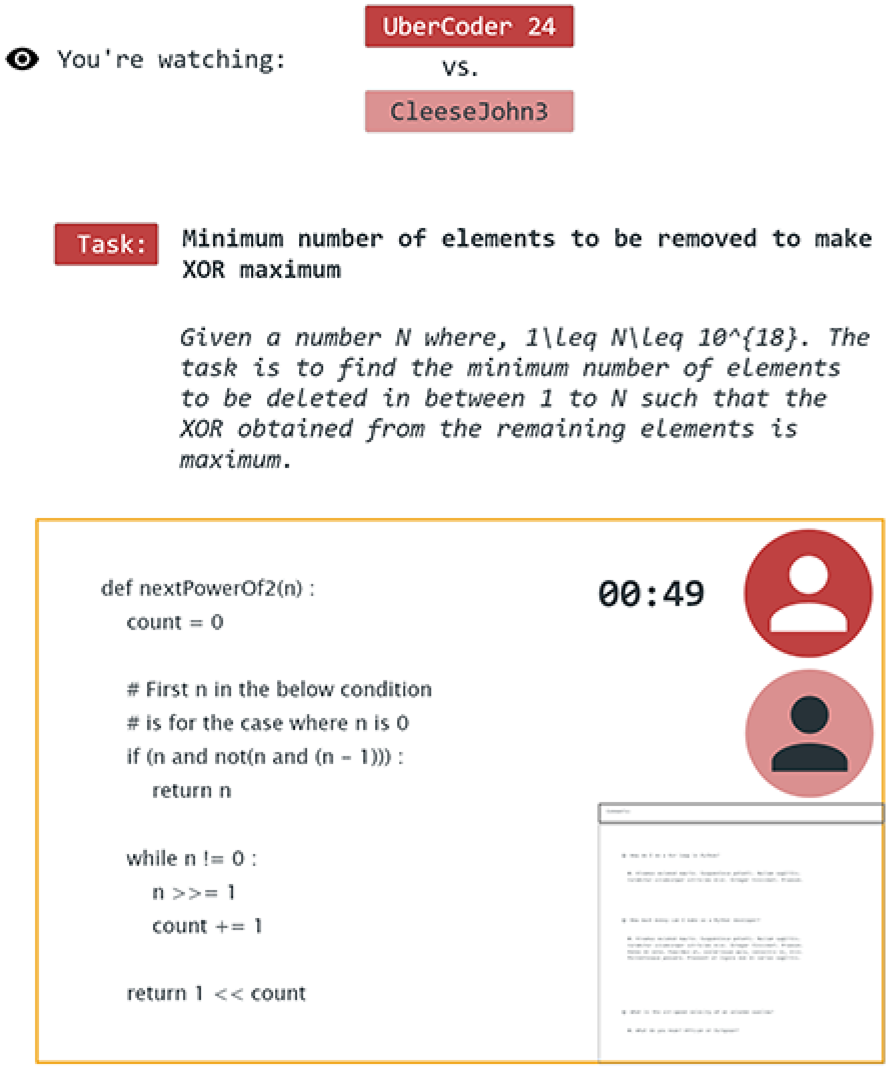
CONTACT

Nullam dictum felis eu pede mollis pretium.

JOBS

Nullam dictum felis eu pede mollis pretium.

v. View an instructional and competitive stream of their chosen programming language.



You're watching: UberCoder 24 vs. CleeseJohn3

Task: **Minimum number of elements to be removed to make XOR maximum**

Given a number N where, 1\leq N\leq 10^{18}. The task is to find the minimum number of elements to be deleted in between 1 to N such that the XOR obtained from the remaining elements is maximum.

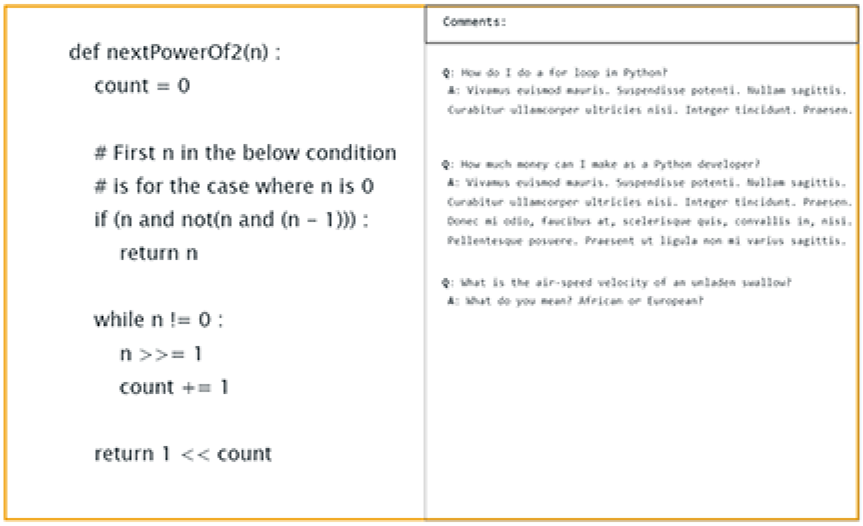
```
def nextPowerOf2(n) :
    count = 0

    # First n in the below condition
    # is for the case where n is 0
    if (n and not(n and (n - 1))) :
        return n

    while n != 0 :
        n >>= 1
        count += 1

    return 1 << count
```

vi. Find specific comments in the comment section of each stream.



```
def nextPowerOf2(n) :
    count = 0

    # First n in the below condition
    # is for the case where n is 0
    if (n and not(n and (n - 1))) :
        return n

    while n != 0 :
        n >>= 1
        count += 1

    return 1 << count
```

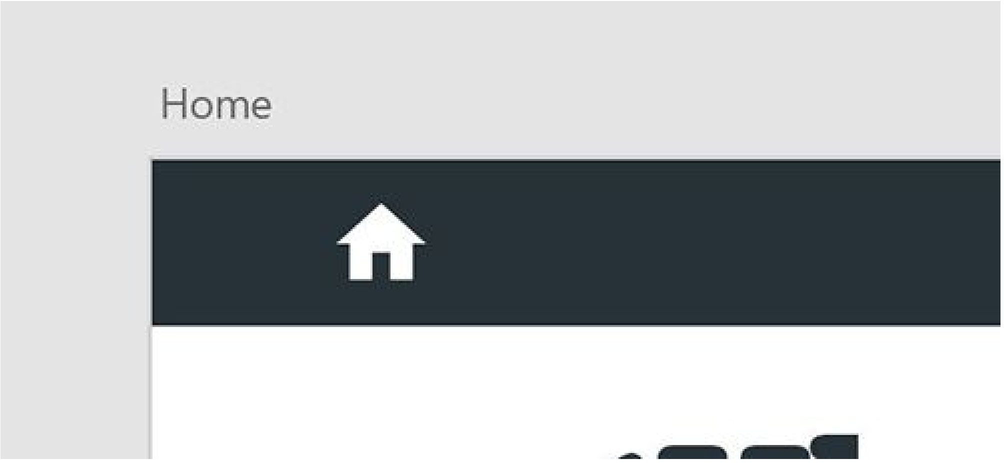
Comments:

Q: How do I do a for loop in Python?
A: Vivamus euismod mauris. Suspendisse potenti. Nullam sagittis. Curabitur ullamcorper ultricies nisi. Integer tincidunt. Praesen.

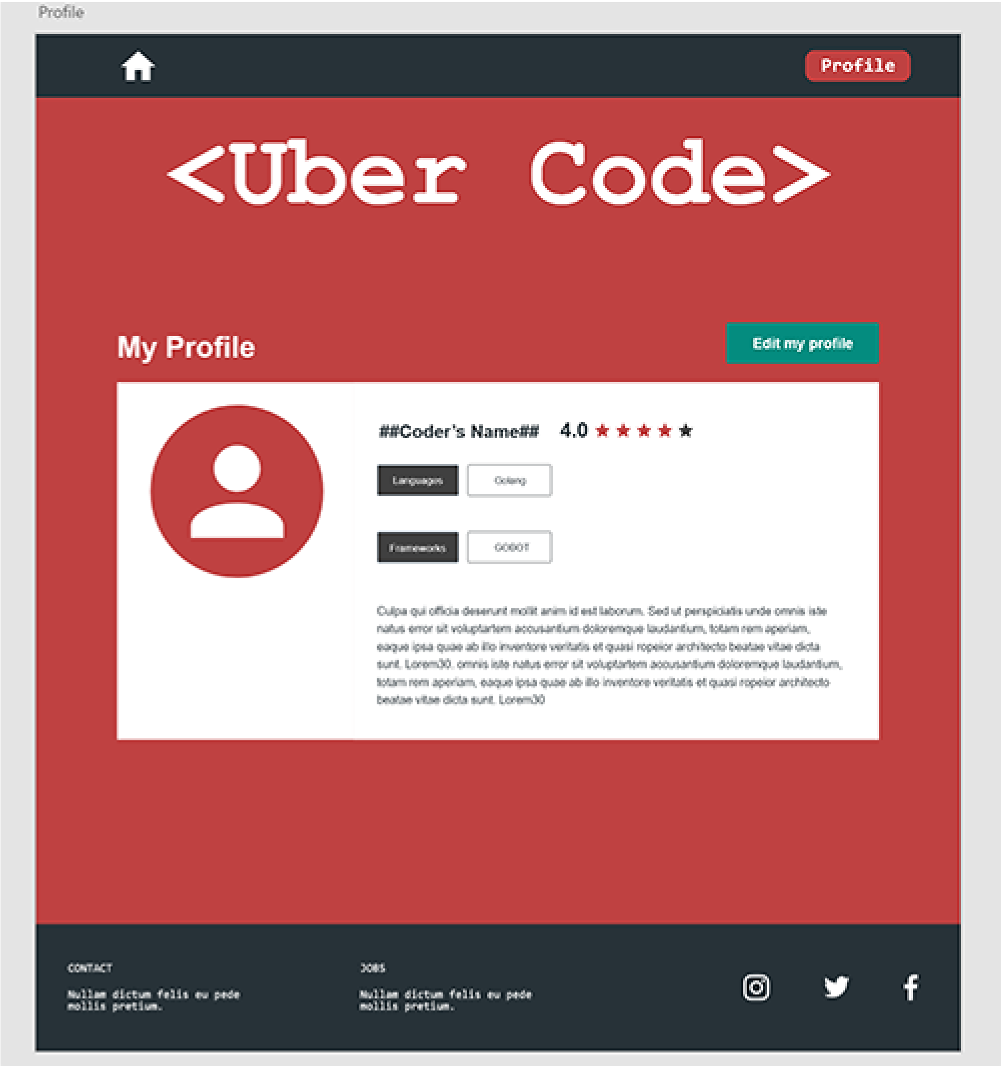
Q: How much money can I make as a Python developer?
A: Vivamus euismod mauris. Suspendisse potenti. Nullam sagittis. Curabitur ullamcorper ultricies nisi. Integer tincidunt. Praesen. Donec mi odio, faucibus at, scelerisque quis, convallis in, nisi. Pellentesque posuere. Praesent ut ligula non mi varius sagittis.

Q: What is the air-speed velocity of an unladen swallow?
A: What do you mean? African or European?

vii. Return to the home page at any time.



viii. View their user profile.



##Coder's Name## 4.0 ★★★★★

Culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis unde omnis iste natus error sit voluptartem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi ropeior architecto beatae vitae dicta sunt. Lorem30. omnis iste natus error sit voluptartem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi ropeior architecto beatae vitae dicta sunt. Lorem30

CONTACT

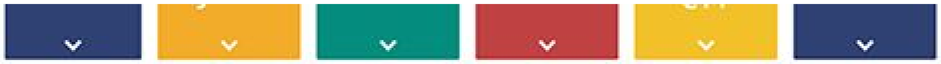
Nullam dictum felis eu pede mollis pretium.

JOBS

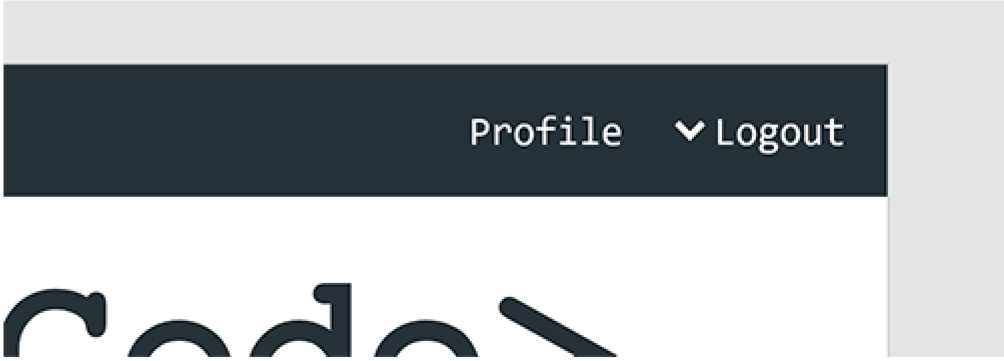
Nullam dictum felis eu pede mollis pretium.

c. Evolution of prototype

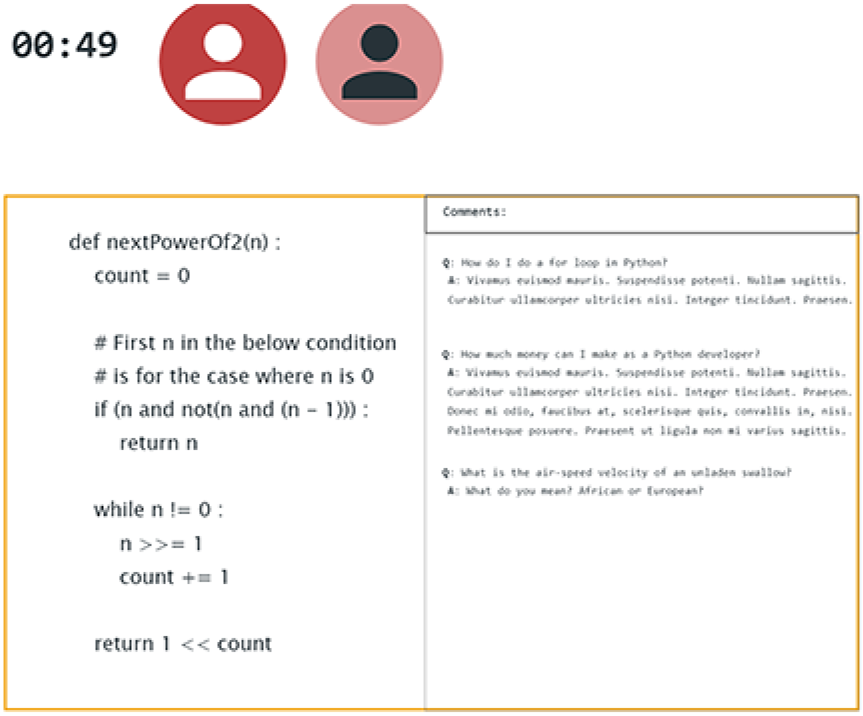
i. Added carats to the dropdowns as it was unclear to some users that the dropdowns were actually dropdowns.



ii. Added a logout button that displayed only when a user was logged in.



iii. Displayed comments in the same row as the coding demonstration/competition to make them easier for users to find.



```
def nextPowerOf2(n) :
    count = 0

    # First n in the below condition
    # is for the case where n is 0
    if (n and not(n and (n - 1))) :
        return n

    while n != 0 :
        n >>= 1
        count += 1

    return 1 << count
```

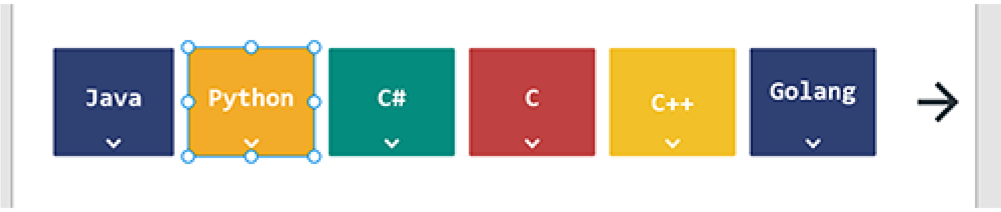
Comments:

Q: How do I do a for loop in Python?
A: Vivamus euismod mauris. Suspendisse potenti. Nullam sagittis. Curabitur ullamcorper ultricies nisi. Integer tincidunt. Praesen.

Q: How much money can I make as a Python developer?
A: Vivamus euismod mauris. Suspendisse potenti. Nullam sagittis. Curabitur ullamcorper ultricies nisi. Integer tincidunt. Praesen. Donec mi odio, faucibus at, scelerisque quis, convallis in, nisi. Pellentesque posuere. Praesent ut ligula non mi varius sagittis.

Q: What is the air-speed velocity of an unladen swallow?
A: What do you mean? African or European?

iv. Added arrow to slide more coding languages into view.



d. **Data Collection**

i. **Controlled Observation -** Throughout the user testing process we gathered qualitative data in the form of controlled observations (notes) about how the user interacted with our interface. To ensure consistency the same testing script was used across all participants in each stage of the user testing process, and the environment (Zoom) remained the same as well. The notes for user testing sessions are located in the testing materials section in the appendix at the bottom of this document.

ii. **Indirect Observation -** Throughout the user testing process we gathered quantitative data through indirect observations. Computer screens as well as the faces and voice of users were recorded during the testing sessions. These captures were subsequently rewatched / relistened to gather valuable feedback that might have been missed during the note taking process. Each tester ranked the subject from 1-5 on the tasks they were asked to complete. 5 meaning completed easily, and 1 meaning completed poorly. This quantitative data can be found in the testing materials section in the appendix at the bottom of this document.

iii. **Questionnaires and Surveys -** In order to gather both qualitative and quantitative data about the usability of the Uber Code prototype users submitted an exit survey after completion of the testing session. A separate survey was administered for each round of user testing in order to extrapolate general trends in usability as our prototype evolved over time. Questions in the survey measured metrics such as overall comfort using the system, ease of use, and the overall likeability of the system in general.

3. Results

a. User Testing Round 1

i. Notes

Mark Lannen - Overall the testing went well, and most participants were able to complete the tasks relatively easily. The main pain points were users not being able to find the home button and not understanding what was meant when they were asked to 'report' on what they were seeing. Also, the live coding section was too large and users often didn't realize that they needed to scroll down past that section to find the comments. All of these issues have been addressed in the updated prototype and script that will be used for the second round of testing.

Addison Boyer - Multiple attempts to press over arrows in the navigation menu, currently doesn't do anything. May be misleading if this arrow doesn't have a function? Scrolling seems to be an issue, i.e. try to condense the content so more is visible without scrolling. It's not apparent when you're on the home screen. One user said, for someone with a mathematical brain tasks are easily completed, but harder for creative artistic thinkers. In other words look out for the left brain and right brain kind of thinker. Wasn't very obvious that it's a home screen, I knew where to look but it wasn't obvious. Wasn't able to identify whether I was logged in, the logout should maybe show a user profile picture (or dropdown to verify that you're logged in).

Colton Gerth - The initial wave of testing went well, although limitations of feedback from the hardware of AdobeXD make some navigation difficult, namely due to the lacking of a scroll bar or feedback of hovering over buttons. Once the users got a feel for how the buttons worked and learned that they were a drop down menu, navigating the web page to view the competitive streams was much more efficient. They also did not find the home page button of <U.C> to be very intuitive.

ii. **Survey**

https://github.com/MarkLannenUM/CSCI443/blob/master/User%20Testing%201.0.pdf

b. User Testing Round 2

i. Notes

Mark Lannen - The second round of testing was smoother as we ironed out several issues that our users ran into on the first round. We added carats to the dropdown menus so that it was more obvious that they were in fat dropdown menus. We also changed visual cues so that it was clearer what state of the app the user was in - i.e., 'logout' was replaced by 'login' if the user was actually logged in.

We also adjusted the testing script as some users were confused by what we meant by 'report what is written…' We changed 'report' to 'read out.'

Addison Boyer - Overall testing seemed to go smoother than last time. It appeared that users were able to recognize the home button easier. One user mentioned that the navigation menu should be condensed into a single component, and shouldn't have an over arrow. One testing participant noticed a typo in the static text in our prototype. The comments section seemed to be more recognizable when condensed closer to the top of the page. One user mentioned that it was difficult to tell whether there was content below or not on small screens. After clicking, I didn't realize the content had popped up. One user mentioned that instead of having next on the last step of user account creation, simply have a button that says finish.

Colton Gerth - The users all found the UI to be self explanatory. They had issues with the scrolling, but those issues are likely due to our improvised screen sharing software from Zoom. There were moments where one of the users was confused by the expanding and minimizing of the comment section, both from not seeing the results right away and by having the screen snap back up. They all liked the registration section. One user noted that the drop down menus could be opened at any time, but could not be closed again.

ii. **Survey**

https://github.com/MarkLannenUM/CSCI443/blob/master/User%20Testing%202.0.pdf

c. Discussion of Results

i. After the two rounds of user testing, it was clear that the UI had a well defined, minimalistic design along with a very pleasing color pallet. However, during the first round of testing, the UI lacked a strong metaphor for the home screen, which caused confusion for the users in their attempts to return to the main page. Not only that, but the UI also lacked proper feedback to the users as they navigated through the coding streams.

The second round of testing featured a new prototype that corrected these issues, and there was a large improvement in user speed and performance through the simple metaphor of an image of a home for the home button, and carrot icons for the stream drop down menus allowed more intuitive navigation.

4. **Conclusion**
    a. **Further Studies**
        i. Uber Code has succeeded in its prototyping development, being able to allow its users to accomplish their given tasks with relative ease. The two phase testing allowed for precise modification to important issues that were rooted out on the first phase, then perfected after the second.
5. **Appendix**
    a. **User Consent Form**
        i. Hello and welcome to Uber Code, an online tool for watching and demonstrating coding skills. Think of it as ‘Twitch’ for coding!

        We would like to take this opportunity to thank you for volunteering your time to help us test this new and exciting web application! I will be reading from this script to ensure consistency between all participants. Please note that Uber Code is still in its development stage and that it may not work under certain circumstances. If that does happen, we will restart the application.

        Understand that the purpose of this exercise is to test our prototype and its usability, not your skills or abilities. If something does not work, the issue is with the developers, not with you. The whole point of this kind of user testing is to discover usability issues with the application, and if we do encounter issues during this testing, we will consider that a positive result as this will help us improve our application.

        If at any time you feel uncomfortable, please inform us and we will terminate the exercise immediately. We will be recording this exercise using video capture of your computer screen, as well as audio recording of verbal user feedback. Your opinions and ideas are important to us. Whenever possible, please try to speak clearly so that the team can document your experience. Do not be concerned about offending us.

        Do you have any questions before we begin?

b. **User Scenarios Form**

i. **Round 1**

1. Scenario: You're interested in signing up for Uber coding and you would like to create an account.

Tasks

- Navigate to the registration page
- Enter in required information, and navigate to the next screen
- Select "Go" as your preferred programming language
- Select "GoBOT" as your preferred framework
- Navigate to your user profile

2. Scenario: Now that you're registered for Uber code you'd like to login and start viewing content.

Tasks

- Login to Uber code
- Navigate and view a Python Stream
- Return to the home screen

3. Scenario: Now that you're comfortable viewing a stream, you'd like to now view a competitive coding lobby.

Tasks

- Navigate to the C# competitive coding lobby.
- Report the users participating in the competition.
- Report the third question in the comments section.
- Choose a programming language of your choice
- Repeat the first 3 steps for the programming language that you chose

ii. **Round 2**

1. Scenario: You're interested in signing up for Uber coding and you would like to create an account.

Tasks (starting on homepage)

- Navigate to the registration page
- Enter in required information, and navigate to the next screen
- Select "Go" as your preferred programming language
- Select "GoBOT" as your preferred framework

- Navigate to your user profile

2. Scenario: Now that you're signed up for Uber Code, you'd like to now view a competitive coding lobby.
Tasks

- Navigate to the GoLang competitive coding lobby.
- Identify the users participating in the competition.
- Read out the third question in the comments section.

3. Scenario: Now that you've viewed some content you'd like to login and start viewing some streams.
Tasks

- Login to Uber code
- Locate the Ruby programming option.
- Next, select and view a Java Stream
- Return to the home screen
- Logout

c. Recordings of Testing Sessions

Round1:
https://umt.box.com/s/w0y2ku7iowmwh30thxkekjb1hu60h1oc

Round 2:
https://umt.box.com/s/i1ipbt9jngk7i1hapwvbw0njemxw8paf

d. Qualitative results of Users Tests

Spreadsheet:
https://github.com/MarkLannenUM/CSCI443/blob/master/UberCode_FinalProject_TestReport.xlsx