Second Iteration:

Team name: Initial Dream
Members: Yifan Li (yl4287), Zixian Zhu (zz2689), Yuqiu gan (yg2684), Shuwan Yao (sy2884)

repo:

https://github.com/ysw1997/COMS4156_InitialDream/tree/main/teamproject%20I2%20new

part 1:

Revise your user stories, from the first iteration and/or project proposal, to reflect what is now fully implemented, tested and working. **Do not include anything that is not actually evident in your github repository and/or you will not be able to show in your final demo.**

As a student to visit this website for the first time, I want to register an account by email, and log in.

#1 Pre-train
As a student to use this system for the first time, I want to finish the pre-training questionnaire related to preventing covid-19 knowledge.

#2.1 Get green daily pass
As a student visiting the campus, I want to submit the form and update my daily health information. If I have no symptom of covid-19 and haven't been to states at risk, I would get a green pass code.

#2.2 Get yellow daily pass
As a student visiting the campus, I want to submit the form and update my daily health information. If I have some symptoms of covid-19 or have been to states at risk, I would get a yellow pass code and not suggested to visit the campus.

#3.1 Upload the test result successfully
As a user who was assigned a yellow pass. I was required to quarantine myself for 14 days. But I got a covid-19 test, and want to upload the result to get a green pass. The website recognised my positive test result. I can request for a green pass again.

#3.2 Upload the test result fail

As a user who was assigned a yellow pass. I was required to quarantine myself for 14 days. But I got a covid-19 test, and want to upload the result to get a green pass. The website can't recognise my test result. It popped out a message telling me to contact the supervisor to apply for a manual review.

#4 Report self quarantine valid
As a user who was assigned a yellow pass. I was required to quarantine myself for 14 days. I don't want to test covid-19, so I decided to quarantine for 14 days. Everyday I reported my location in my quarantine area, so my quarantine is valid.

#4 Report self quarantine valid
As a user who was assigned a yellow pass. I was required to quarantine myself for 14 days. I don't want to test covid-19, so I decided to quarantine for 14 days. One day I got out of my quarantine area, the gps location detected that, and I was notified that this quarantine is not valid now. I need to restart a new 14-days-quarantine.

#5 Receive contact reminder
As a user logged in, I will receive a notification in the home page of numbers of positive cases in the building I visited on the same day.

part 2

Write a test plan that explains the ***equivalence partitions*** and ***boundary conditions*** necessary to unit-test each of the major subroutines in your system (methods or functions, excluding constructors, getters/setters, helpers, etc.) and then implement your plan. Associate the names of your specific test case(s) with the corresponding equivalence partitions and boundaries (if applicable). Your test suite should include test cases from both ***valid*** and ***invalid***equivalence partitions, and just below, at, and just above each equivalence class boundry (or inside vs. outside the equivalence class when boundary analysis does not apply). Note the same test case might apply to multiple equivalence classes. Say there is a method whose input should be an integer between 1 and 12. Then there is an equivalence class 1-12, an equivalence class <1 (or <=0), and an equivalence class >12 (or >= 13). A test case with input 0 would be just below the lower boundary of the equivalence class 1-12 and also at the high boundary of the equivalence class <1.

Include the link to the folder in your github repository that contains your automated test suite.

Boundary conditions:

1. Password should be at least 6 characters

If I fill a right formatted username and a 5 characters password, it will flash ‘Password should be at least 6 characters’ error.

If I fill a right formatted username and a 6 characters password, it will proceed.

If I fill a right formatted username and a 6 characters password, it will proceed.

Equivalence partitions:

If I fill a not email format username, it will flash ‘The email address is badly formatted.’ error.

If I fill a not email format username, it will flash ‘The email address is already in use by another account’ error.

Part of test plan explanation:

After I login, If I fill in the correct username and choose today’s today, and ‘0’ for the symptoms, select ‘NWC’ for visited building, ‘NewYork’ for visited state in the past 14 days.I will jump to the green pass page’, which says ‘green pass’, and a link to get back to the index.
If I fill in the correct username and choose today’s today, and select symptoms, ‘NWC’ for visited building, ‘NewYork’ for visited state in the past 14 days.I will jump to the yellow pass page’, which says ‘yellow pass’, and a link to report your self quarantine, a link to upload test result and a link to get back to the index.
All the steps mentioned above are tested by unitest, from the website and postman. Not all the unitest are included in the explanation.

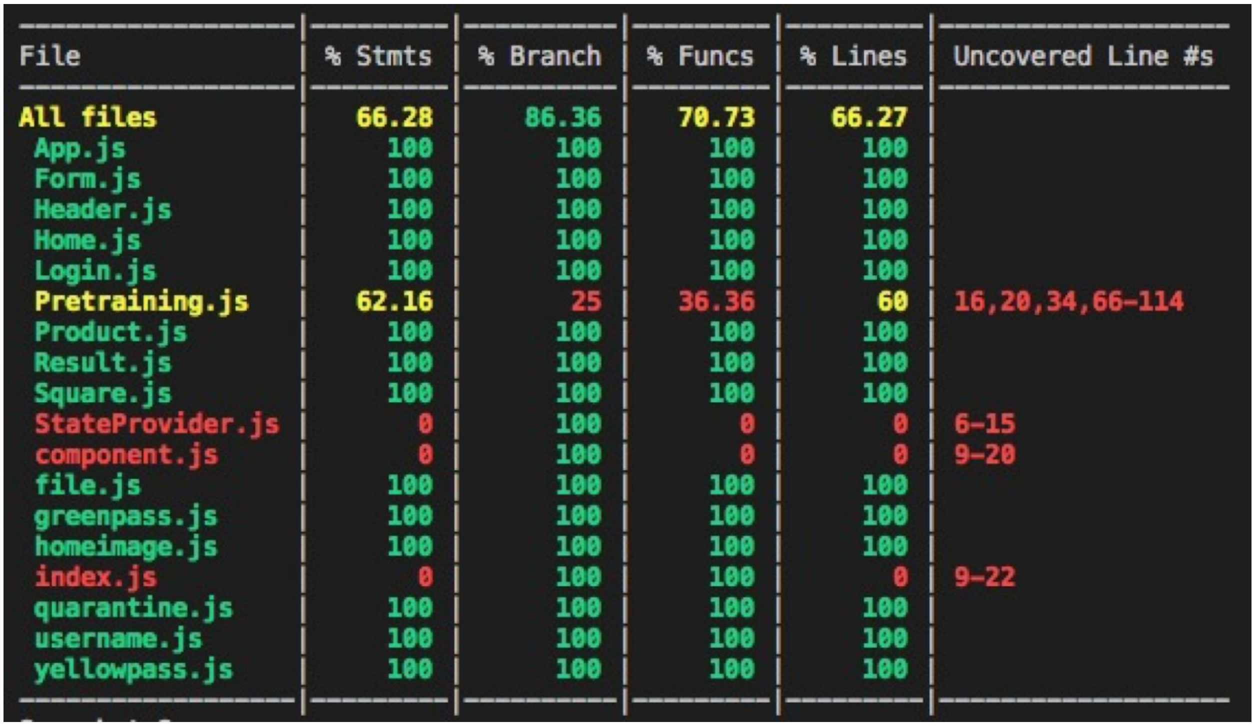
part 3:

Measure the ***branch coverage*** achieved by your automated test suite. This requires using a coverage tool appropriate for your programming language and platform. Branch coverage should strive to achieve 100%, but may not reach 100%. Add more test cases until you reach at least 90%. Each additional test case should try to force a particular branch that was not previously covered. Tell us what branch coverage you finally did achieve. If the coverage tool reports less than 100% (that is, between 90% and 99%), discuss one example of a branch that your test cases did not cover and explain why it is difficult to test this branch. If you were unable to reach 90%, explain why not.

Include the link to the folder in your repository that contains the coverage test reports. Note this means you need to configure your coverage tool to produce reports that can be saved as files in your repository.

Frontend Text Coverage Report:

https://github.com/ysw1997/COMS4156_InitialDream/blob/main/Submissions/Coverage%20report%20for%20frontend.docx



```
------------------|---------|----------|---------|---------|-------------------
File              | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
------------------|---------|----------|---------|---------|-------------------
All files         |   66.28 |    86.36 |   70.73 |   66.27 |
 App.js           |     100 |      100 |     100 |     100 |
 Form.js          |     100 |      100 |     100 |     100 |
 Header.js        |     100 |      100 |     100 |     100 |
 Home.js          |     100 |      100 |     100 |     100 |
 Login.js         |     100 |      100 |     100 |     100 |
 Pretraining.js   |   62.16 |       25 |   36.36 |      60 | 16,20,34,66-114
 Product.js       |     100 |      100 |     100 |     100 |
 Result.js        |     100 |      100 |     100 |     100 |
 Square.js        |     100 |      100 |     100 |     100 |
 StateProvider.js |       0 |      100 |       0 |       0 | 6-15
 component.js     |       0 |      100 |       0 |       0 | 9-20
 file.js          |     100 |      100 |     100 |     100 |
 greenpass.js     |     100 |      100 |     100 |     100 |
 homeimage.js     |     100 |      100 |     100 |     100 |
 index.js         |       0 |      100 |     100 |       0 | 9-22
 quarantine.js    |     100 |      100 |     100 |     100 |
 username.js      |     100 |      100 |     100 |     100 |
 yellowpass.js    |     100 |      100 |     100 |     100 |
------------------|---------|----------|---------|---------|-------------------
```

Explanations for why the branch coverage is under 90%:
Because we can’t test all situations in pretraining.js file, some functions in this file we never need in frontend, which means that they don’t show on the webpage, If I comment them, there will be no difference in the front end part and the branch coverage will turn to 100%, but the backend need these functions, and I search with google how to test this part of “fake code” and get no idea.

Frontend Checkstyle Report:

https://github.com/ysw1997/COMS4156_InitialDream/blob/main/Submissions/frontend%20checkstyle%20report.docx

Backend Checkstyle Report:

https://github.com/ysw1997/COMS4156_InitialDream/blob/main/Submissions/pylint_report.rtf

https://github.com/ysw1997/COMS4156_InitialDream/blob/main/Submissions/flake8.rtf

part 4:

As part of the release effort, you need to institute ***continuous integration*** for your codebase. The idea is to integrate automated build and test with your version control repository using Travis CI or a similar tool, so that build and test is automatically initiated whenever there is a new commit to the main branch of your github repository. Make sure to ask for help far in advance of the assignment deadline if you have trouble getting CI to work.

Include links to the files in your repository that configure the CI tool(s) and a folder in your repository that includes the CI reports. Note this means you need to configure your CI tool to produce reports that can be saved as files in your repository.

https://github.com/ysw1997/COMS4156_InitialDream/tree/main/.github/workflows