**PART A**

**1. Associate a README.TXT with the app outlining technical implementation matters**

Readme.txt file is included in the Git repository.

**2. Write HTML help document and associated? icon covering common user interface elements in the UI**

**3. Write a Roadmap going forward of the developments to the interface, including in development repository where code is stored**

**4. Move 3rd party dependencies to CDN repositories, remove 3rd party files from the codebase**

**5. Why is it essential to implement the above (1-4?)**

**6. What other types of documentation may be necessary for this project?**

At the completion of the project, a final report on what changed from the original proposal through to the final completed app would need to be completed. Not only to detail what has changed but also to inform the client for the project what has been implemented and what may need modifications or updates in the future.

**PART B – Continuous Improvement**

**7. What portions of the development went particularly well**

I excelled in the HTML and CSS, so that went pretty smoothly in terms of creating the mock-up design for the site, and further implementing it into the completed version of the app. I was also able to get some of the PHP functions such as my join, login and register functions written and functional before needing help with the base case to get them to function correctly.

**8. What was the most difficult to implement**

The most challenging part of this development was the building the base case (as it was a completely new style of coding for me and it was a steep learning curve to make it work and then another steep learning curve to relearn and reformat it to make it functionally correct) as well as the JS section of the coding (it is my weakest coding language and I still have a long way to go when it comes to being fully competent).

I have also found that React has been a challenge for me to learn and implement. I completed the React Tic Tac Toe exercise twice but still struggled with grasping basic concepts of the technology as a whole.

**9. If you had the chance to do this again, what would you do differently?**

Get started on learning my framework at an earlier time. React is by far the most challenging part of the assessment even bypassing the JS and PHP integration that I implemented in Proj 2 and Ux2. I would also spend more time in the early days of the assignment working on my code instead of putting it as a lower priority than the other assessments I had due. Putting it off resulted in a massive time crunch during the midsemester break and more than one hit to my mental health when my code wasn't complete by the due date.

**10. What parts of the implementation incomplete at this stage of delivery?**

At this point of delivery, there are 3 critical parts of my original project plan that are still outstanding and incomplete.

1- The ability to access the app with a QR code

2- The mobile phone (text) and push notification code

3- The restriction on having a user register with the same costume more than once per day.

**11. Write and reflect on "Quality Assurance" how are you practising this?**

Within web app, development and ICT solutions in general quality assurance is one of the most important elements to implement. Quality Assurance is a series of checks, balances and requirements that ensure that the resulting app that is created by a developer meets the expectations and requirements of the initial proposal and of the client.

To ensure that as many (if not all) of the required deliverables for The Cosplay Queue are met, I have been referring to the proposal created in Proj 1 at every step of the development process. If I have made changes to the scope of the project or to one of the required deliverables, I have documented it and discussed reasons as to why they haven't/can't be implemented at this time.

Changes such as the removal of the restrictions limiting a user to a single registration per costume per day have been decided on via discussion with peers and potential users of the application.

Upon completion of each stage of development, a set of checks were performed against the list of required functionalities to ensure that everything was working as expected.

**12. How much of the prototype UX1 remains in the final project?**

Looking back at the prototype that I created in UX1, the majority of the home page design has remained as it was in the original design. The contents of the other pages that were submitted still remain; however, their location has changed so that they now reside solely on the home page, contained within collapsible sections and modals.

**13. Where has your project Object-Oriented programming implemented?**

Object-Orientated Programming (OOP) was implemented in multiple ways throughout the project, but most prominently through the use of PHP and JavaScript as key coding languages.

The implementation of the base case is an example of the "Don't Repeat Yourself" (DRY) principle as it demonstrates a central location for all the code and from there functions can be called as many times as necessary without repeating code.

Using React to construct the backend of the app is also an instance of OOP as the way JSX is structured as a language, both data and functions are passed at the same time and within the same object.

**PART C – Presentation**

**14. Student to present three measurable criteria from the project plan (PROJ1)**

3 measurables: Join Queue, Register, Dequeue, Login, Update Details

**15. Present Web App (solution) to class**

Complete

**16. Seek feedback from your peers on level of quality, based on project plan**

Complete

**17. Rectify any failings as a result of this activity**

Complete

**PART D**

**1. Check final application in a Desktop version of Chrome & Firefox + IOS and Android phones**

**2. Bring UX2, PROJ2 & PROJ3 together in one .zip, write a README describing installation for operations staff**

**3. Include in the README all the technologies used in the app, places where they were used and versions you recommend**

Readme.txt file is included in the Git repository.

**4. Confirm functionality in relation to plan PROJ1 highlight areas that changed or were not implemented.**

The initial database design has been completely overhauled to remove duplicate data stores and make it more streamlined.

**Completed and operational functions ->**

* Need to be able to register users for new accounts as well as access existing accounts.
* The admin user needs to be able to preview the queue at any time in order to prep the lighting rig.
* Only the admin users can dequeue anyone from the queue, while individual users can only dequeue themselves.
* Photos need to be easily pulled from the database along with the associated data to help with cosplayer identification and photo processing.
* Cosplayers need to be able to dequeue themselves whether registered users or unregistered users.
* Cosplayers need to be able to update their details that are stored in the database at will. Within the cosplay community, people's identities such as names, genders and screen names are fluid, so they change regularly.
* The admin should clear event table after each group of events is completed and the photos processed. - Complete and in the admin panel

**Altered from initial business rules ->**

* Needs to be able to alert the cosplayer using popup notifications, email and/or text which also needs to happen automatically. -> Currently not implemented as there was a cost incurred with this functionality. In its place is a live update feed that shows where the user is in the queue that updated every 60 seconds.
* Needs to be able to stop a user from queuing up with the same costume more than once per day but allow for multiple costume changes during the day. -> Upon reflection and discussion with other members of the cosplay community this part of the project scope is unneeded and unnecessary as it prevents users re-joining the queue after dequeuing, and also prevents newly established groups that joined together after the initial photoshoot from being photographed together.

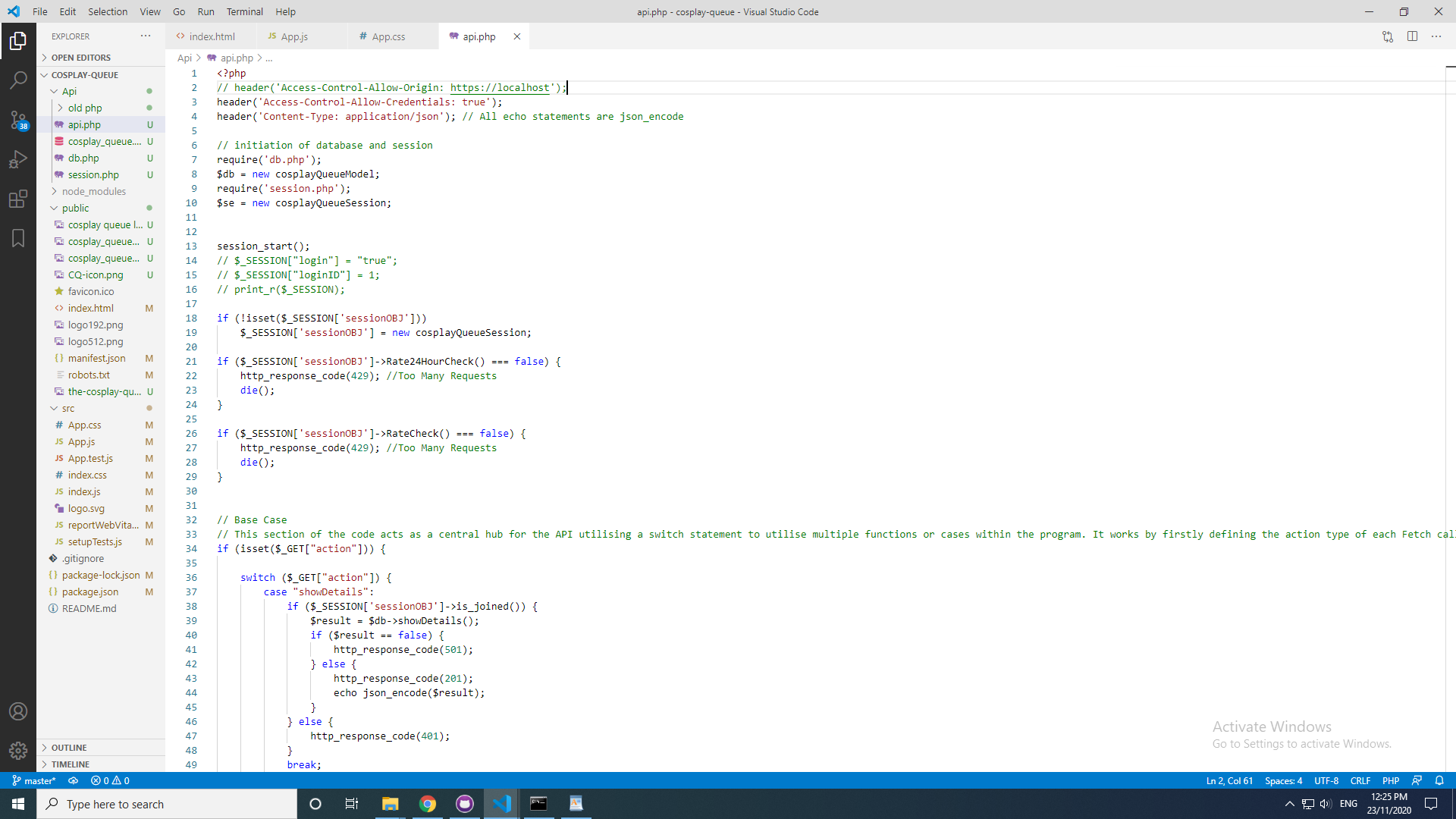
**Currently Incomplete ->**

* Must be able to utilise QR codes to enable the non-registered user -> currently not implemented but will be implemented at a later state
* The form needs to autofill with cosplayer information from the database if they are signed in. -> Not complete at this point in time but autofill is present within the app.

**PART E – SECURITY AUDIT**

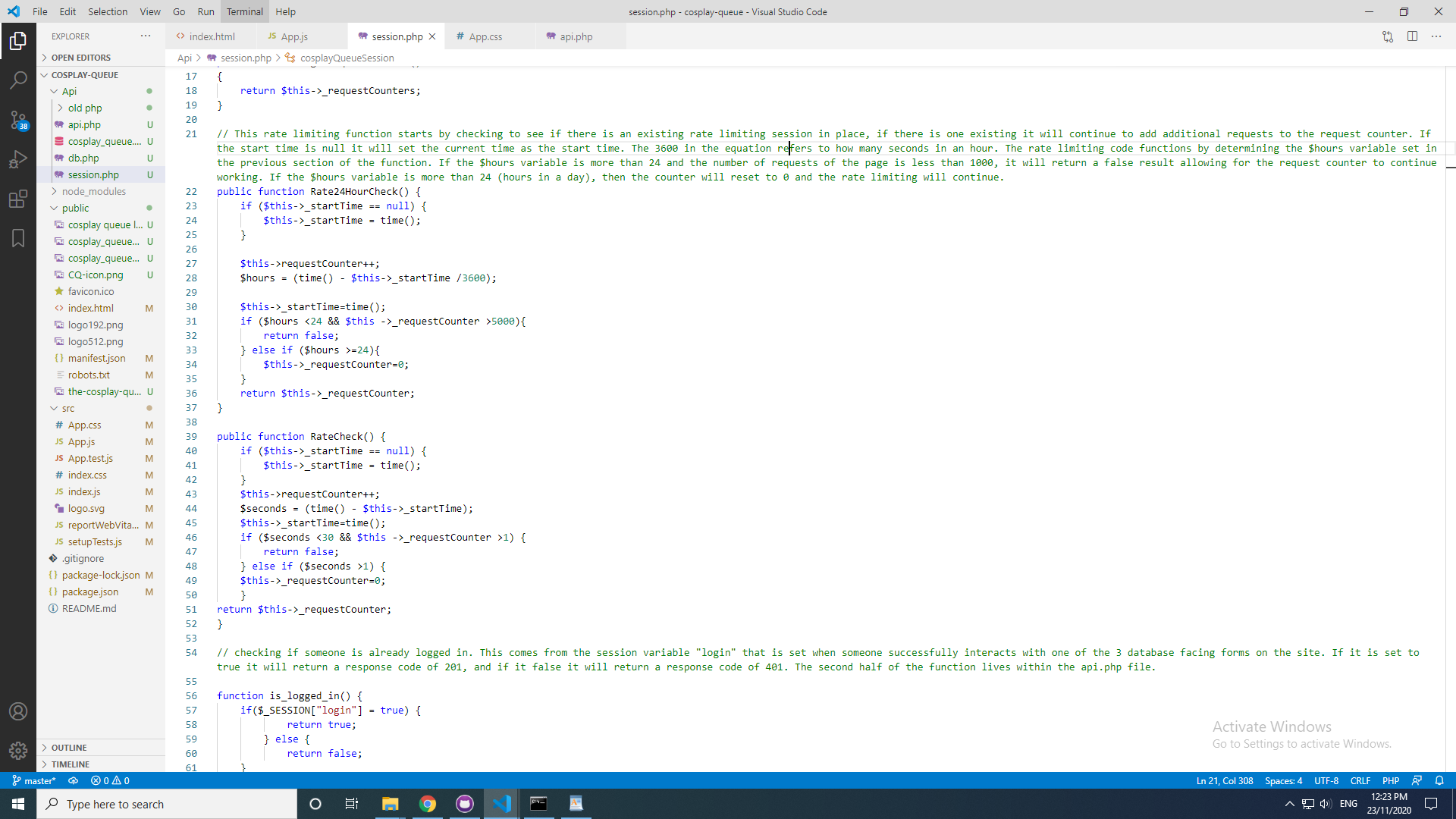
**6. Secure app by resetting/removing admin passwords**

**7. Remove localhost from referrer security, add hosting domain referrer**

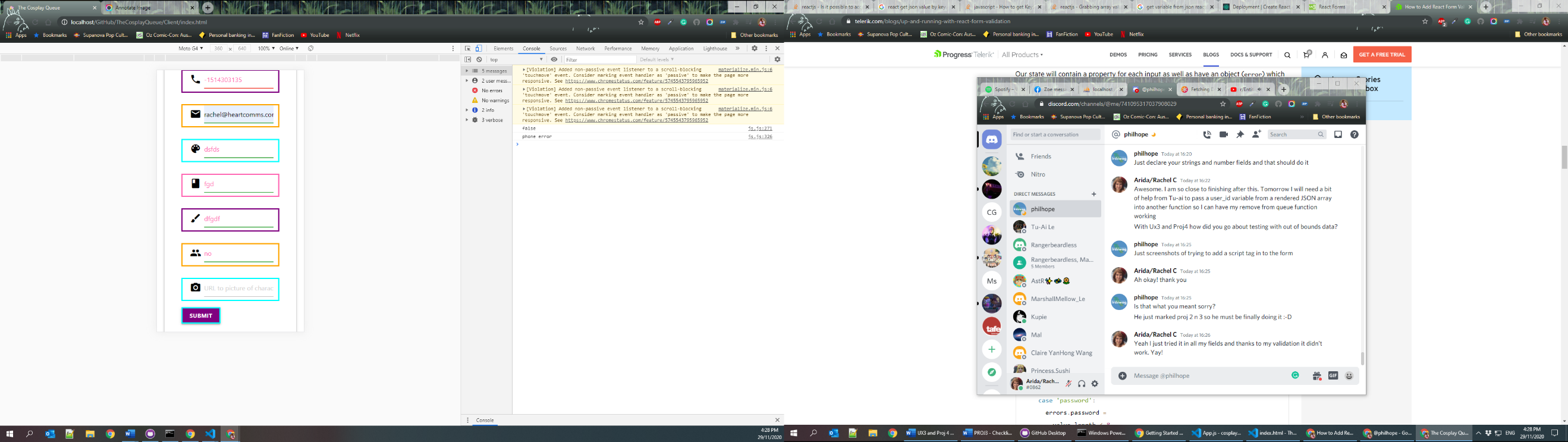
****

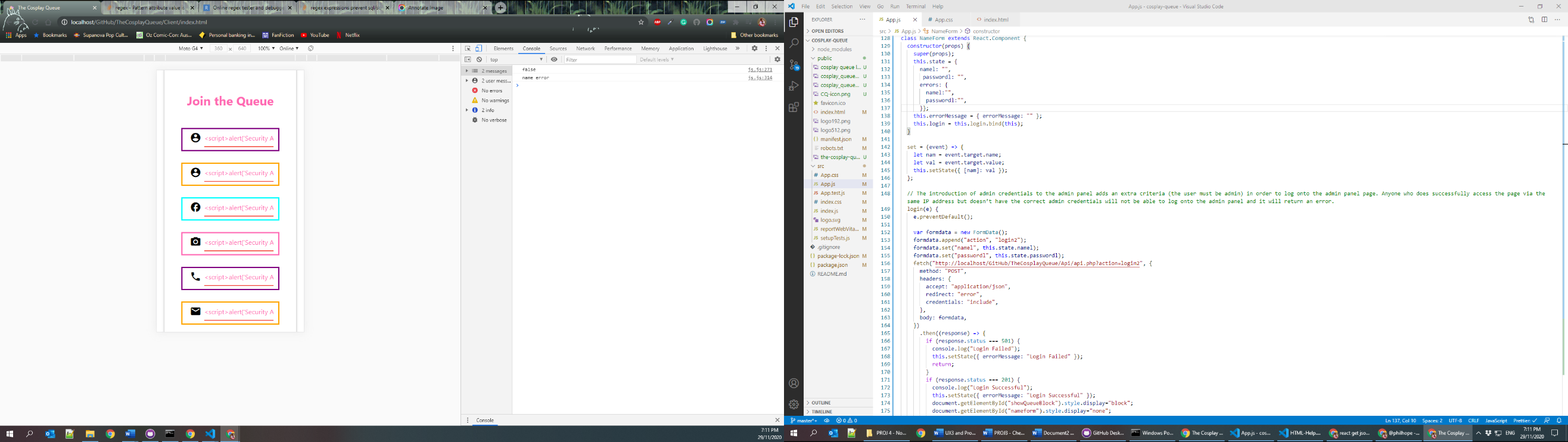
**8. Remove log data & test data from app**

**9. Reset rate limits to once every 30 seconds & a maximum of 5000 requests (permission given to change 500 to 5000 to accommodate for auto refresh on app)**

****

**10. Test with out of bounds data like negative numbers, Inject <script>alert(‘Security Alert’);</script> Update or delete on IDs that don’t exist. Screenshot results and suggest a remedy**

Negative number – blocked by form Validation. No remedy needed.

Script Injection – blocked by form Validation on all fields. No remedy needed

Update or delete on IDs that don’t exist. ID’s are determined in the backend of the code and set using php Sessions. These cannot be accessed via the front end as they are generated during the process of logging in/registering/joining the queue. No remedy needed.