**Fillr / Rakuten QA Assignment**

**Task 1**: General Testing

* Since, the new sign-in page is a replacement to the old legacy page, there would be lots of users existing already. So, we should be testing if the new sign-in page is able to handle the expected load by performing a **load test**.
* A **performance testing** should be done as we are dealing with a page where we would need some response from the DB if the sign-in credentials entered are correct or not. So, we should be checking the response time and stability of the application as well.
* To reduce the risk from web application attacks like XSS attack and SQL injections, we should consider having a **security testing** to not let our customer data be at risk and the organization too.
* As the page is a React Native response page, we should be testing the application in both Android and IOS platforms if the application is supported or facing any compatibility issues.
* With a minimum knowledge of having API testing, I would suggest one to test API responses to see if the endpoint is able to communicate with the respective DB to retrieve the data requested with the parameters given. This also verifies the mapping between the application and DB.
* **UI and Functional testing** should be included to verify cosmetic issues and workflow of the features by including **UAT** (User Acceptance Testing) and **UET** (User Experience Testing)
  + UI testing can be automated by comparing the golden screenshots with the current test screenshots in-order to identify any changes respectively e.g., any cosmetic issues like icon’s font or colour change and even position change which leads to overlapping or breaks in the text.
  + Functional testing can also be automated by having multiple test data for all the scenarios mentioned below
* A test plan can have multiple features covered among which Login page comes under a single feature with different scenarios / cases covered.
* DB testing can also be done as per the given information, the API talks to DB. We should be testing if the SQL queries written are giving the expected output and then we can have the connection between the API & DB. To test DB we can create a test data with all possible scenarios and test accordingly.

Please find below for sample scenarios covered which includes positive and negative testing scenarios and the format that I usually follow while preparing a test case or scenario in any kind of app either Web or Mobile. Only the steps change but format remains same.

Please click [here](file:///C:\Users\sagar\OneDrive\Documents\Fillr_QA_Assessment.xlsx) to find scenarios.



**Note**: The test cases written by considering the Login page is part of a native app. However, if we need it for the web app then there would few other scenarios adding to verify on different browsers, URL and so on.