React Native Front-end –

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | **Date** | **Solution** | **Comments** |
| Dark colour scheme not being detected on mobile. | 12/12/21 | Added "userInterfaceStyle": "automatic" to app.js. |  |
| Expo SDK version upgrade not completing. Due to expo initially being installed via npm as root. | 24/12/21 | Wipe the old expo install from the system. Npm uninstall was not sufficient, the files needed to be deleted from usr/bin aswell. Fresh install of Expo. | This took some time to recifiy as it was initailly unclear that npm uninstall wasnt removing the binaries for expo. |
| Cant run expo install. | 24/12/21 | sdkVersion needed to be specified in the app.json file. |  |
| On mobile the notification bar overlaps some content. | 28/12/21 | The SafeAreaView and SafeAreaProvider components needed to be used in replacement of the view components. | Thesse new components are provided by react navigation. |
| ScrollView wont scroll on mobile. | 28/12/21 | ScrollView needed to be wrapped inside a regular View with a style of flex:1. |  |

FLASK API -

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | **Date** | **Solution** | **Comments** |
| sqlalchemy.exc.OperationalError: (MySQLdb.\_exceptions.OperationalError) (2005, "Unknown server host '@localhost' (-2 )")  Couldn’t connect to the db with the flask account.  The flask account had a randomly generated password which contained an @ symbol as the last character, this confused SQLAlchemy into thinking the sepeartor for the USER PASSWORD and SERVER ADDRESS was sooner than it actually was, ticking sqlaclhemy into think the server address was @localhost instead of just localhost. | 03-01-22 | Generate a new random password which doesn’t contain an @ symbol. |  |
| JS fetch request to the flask API failed due to a failed prefight CORS policy OPTIONS request to the api. | 26/01/22 | @app.after\_request  def after\_request(response):      header = response.headers      header['Access-Control-Allow-Origin'] = '\*'      header['Access-Control-Allow-Headers'] = 'Content-Type,Authorization'      return response  Generate a response that is sent after a request to the API using the @app.after\_request decorator, sending back the allowed origins and headers for the api. |  |