BLH Beta App Documentation

1. Description: The app is designed to allow users to browse BLH’s catalogue of wigs and products, as well as “try them on” using AR/deepfake technology
2. Progress:
   1. Overview: Currently, the app has been developed to point where core functionalities, as well as surface appearance for the app has been completed. Next steps include implementing peripheral functionalities such as secondary buttons, as well as integrating dynamic values to allow for pages to sync with database of values.
   2. What is done:
      1. Visual layout and buttons linking pages shown in photoshop files
      2. Core camera function including taking pictures and videos
      3. Displaying video and sending video to cloud server
   3. To be completed:
      1. User profile and authentication
      2. Implementing functionality for peripheral buttons
      3. Cause displayed items to be synced with external values
      4. Allow for app to pull data and files in real time from external sources
3. Notes regarding code:
   1. The app is written primarily using Flutter, which uses the Dart coding language, which is similar to C# and Java.
   2. Each page has it’s own class, which returns a widget which contains the components of the page.
   3. Most page classes inherit from the StatelessWidget class, which dictates that the page does not change over time, which means they will be more or less static
   4. A few page classes inherit from the StatefulWidget class, the counterpart to stateless widgets, and these pages will change over time.
   5. Some pages return Scaffold as the main widget, scaffolds are used when a top bar/bottom bar need to be implemented, as Scaffold comes with fields for both options
   6. Some pages return Container as the main widget, containers on their own do not display anything, but are useful for holding things, as the name implies. They are primarily used in the app for sizing, adding background colour/image, as well as adding padding between other widgets.
   7. Rows, Columns, as well as Stack widgets are also very common in the code. As their names imply, they are used to put widgets in a row, column, or on top of one another, respectively.
   8. The const keyword appears quite frequently in the code, and it means that the widget/item it is attached to will be constant, thus if a field is replaced with a variable value, the const keyword is no longer applicable and will break.
   9. Regarding the upload function, currently the upload link is set to http://(some IP)/C:/Images/SaveFile.php, the IP will need to be changed according to what the IP of the cloud server is to work.
   10. Images and fonts are pulled from the assets folder, which can be found in the ar\_beta/assets folder
   11. Most of the camera code uses the camera package, and is pulled from the example found here: <https://pub.dev/packages/camera/example>
   12. Whenever adding new packages, they need to be added at the top, as well as in the pubspec.yaml file under dependencies

Additional Resources:

Flutter API reference Documentation: <https://api.flutter.dev/>

Package and Plugin Repository (usually has documentation on various used packages): <https://pub.dev/>