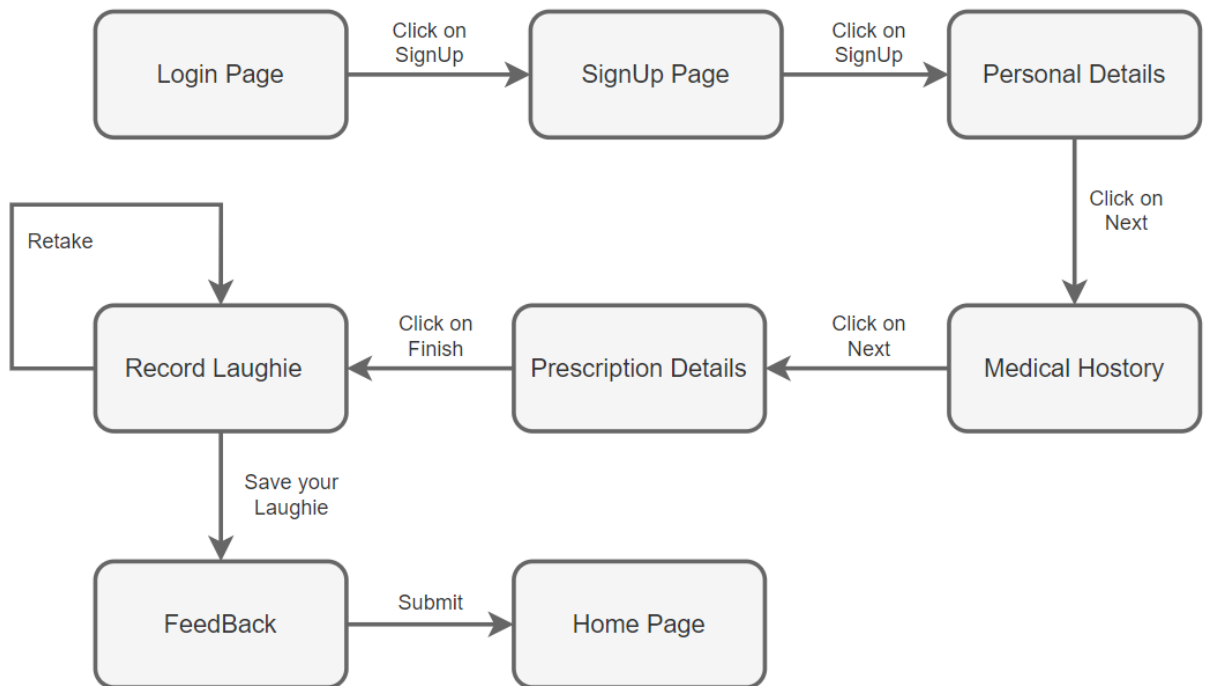


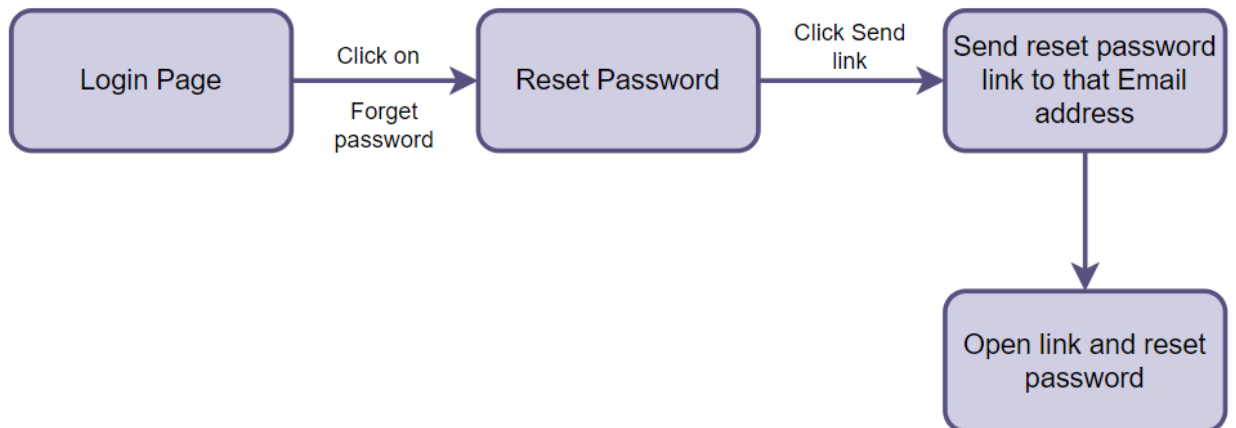
# Laughie Documentation

## 1. Workflow:

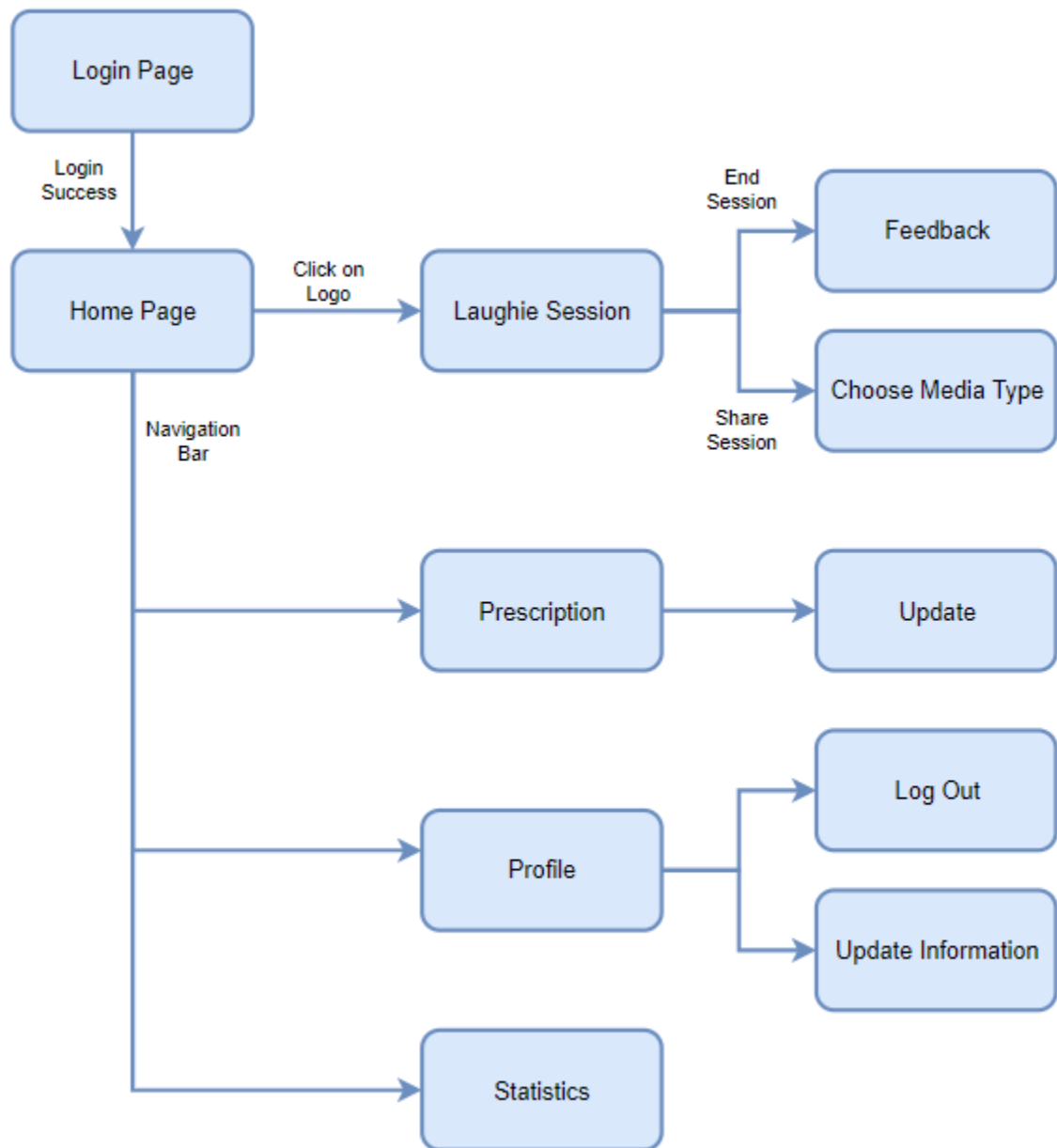
### a. New User:-



### b. Reset Password:-



c. Logged-in User:-



## 2. Coding Standards/File Structure:

### 3.1 Naming conventions

The following three types of identifiers are to be used:

1. **UpperCamelCase** names capitalize the first letter of each word, including the first.
2. **lowerCamelCase** names capitalize the first letter of each word, except the first which is always lowercase, even if it's an acronym.
3. **lowercase\_with\_underscores** names use only lowercase letters, even for acronyms, and separate words with \_.

For a detailed guide on the naming, do refer to the [dart styling guide](#) before starting with the coding.

### 3.2 Indentation and module separation

1. Indentation

Flutter provides [automatic formatting](#) for the code on various IDEs. Setup the automatic formatting before starting to code.

2. Module separation

Use the following separation protocols for the code files.

**/screens/myCode.dart** : Add all the front-end files here

**/rewidgets/myCode.dart** : Add all the reusable components here

**/services/myCode.dart** : Add all backend services files here

**/models/myCode.dart** : Add all the machine learning model files here

**/helpers/myCode.dart** : Add all the helpers file here (camera, tflite etc)

**/utilities/myCode.dart** : Add all the API keys here

**/constants/myCode.dart** : Add all the constants here

**/assets** : Add all assets like fonts, images here

## 3. Firebase structure:

[\[API documentation\]](#)

### 4.1 Firebase Authentication

Firebase Auth is used in this application for user authentication. It authenticates using email id and password.

## 4.2 Firestore Database [NoSQL]

Firestore Database is used to store all the user related data. There is a single collection of “users” that contains different documents for each user and all the documents have further nested fields and collections. These nested collections are used to track the user progress while the fields are used to store the user data.

Sample photos:


The screenshot displays the Google Cloud Firestore console interface. The breadcrumb navigation at the top indicates the path: `users > 18sJ2WVEMWPORTun7aQdWh2B1eh2`. The interface is divided into three main panes:

- Left Pane:** Shows the 'users' collection. A document with ID `18sJ2WVEMWPORTun7aQdWh2B1eh2` is selected.
- Middle Pane:** Displays the fields of the selected document. It includes a nested collection named 'sessions'. The fields shown are:
  - `city: "Ahmedabad"`
  - `contact_number: "9099695255"`
  - `country: "India"`
  - `date_of_birth: 29 October 2000 at 00:00:00 UTC+5:30`
  - `email: "nilaygaandhi@gmail.com"`
  - `ending_date: 31 August 2021 at 00:00:00 UTC+5:30`
  - `filePath: "file:///storage/emulated/0/Pictures/1628614972607.mp4"`
  - `gender: "Male"`
  - `has_medical_history: false`
  - `has_recorded_laughie: true`
- Right Pane:** Shows the 'sessions' collection. Two documents are listed:
  - Document ID `10082021` with fields:
    - `date: "10082021"`
    - `session_data` (array):
      - Index 0: `q1: 4`, `q2: 5`, `time: 10 August 2021 at 22:32:56 UTC+5:30`
      - Index 1: `q1: 4`, `q2: 5`, `time: 10 August 2021 at 22:34:28 UTC+5:30`

## 4.3 Firebase Storage

Firebase storage is used for storing the user's profile photos and also storing the Laughie tutorial videos.

Sample photo:



The screenshot shows the Firebase Storage console interface. At the top, there is a URL bar with 'gs://laughie-52bd5.appspot.com' and an 'Upload file' button. Below this is a table listing the contents of the storage bucket. The table has four columns: 'Name', 'Size', 'Type', and 'Last modified'. The first row is a folder named 'profile\_pictures/'. The second row is a video file named 'laugh\_along.mp4' with a size of 2.37 MB, type 'video/mp4', and a last modified date of '10 Aug 2021'. The third row is another video file named 'record\_laughie.mp4' with a size of 4.2 MB, type 'video/mp4', and a last modified date of '10 Aug 2021'.

<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	profile_pictures/	—	Folder	—
<input type="checkbox"/>	laugh_along.mp4	2.37 MB	video/mp4	10 Aug 2021
<input type="checkbox"/>	record_laughie.mp4	4.2 MB	video/mp4	10 Aug 2021

## 4. CI/CD using GitHub Actions:

### 5.1 How to get started?

Tool recommended: [GitHub Desktop](#)

Steps:

1. Clone the repository.
2. Create your own branch with proper naming convention.
3. Commit all your changes on your branch and once you are done with implementing the functionality, create a pull request.
4. Move to the integration process.

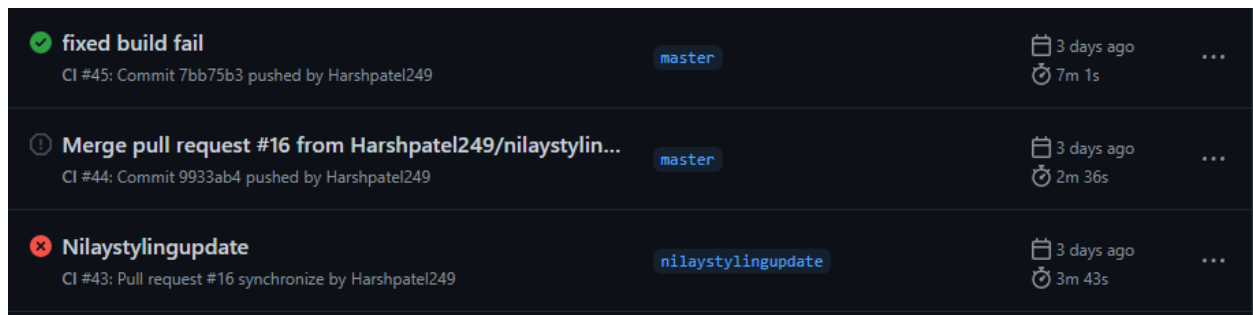
### 5.2 Integration process

The integration process requires the changes to pass the GitHub Actions build test and the following are the steps for the same:

Steps:

1. Create a pull request.
2. Resolve all the conflicts with the master branch and ask a peer to review the solutions so no content is lost.
3. After resolving the conflicts, create a merge request.
4. Wait for the GitHub Actions to test the changes triggered on the creation on merge request.
5. If the build test is passing, merge the changes.
6. If the build test fails, check the logs in the Actions section and resolve the errors and go to step 1 again.

Example screenshot of the Github Action run:



## 6. Features Overview:

Following are the main features of the laughie app:

1. User account [create, update and statistics].
2. Record laughie [Audio or video].
3. Use the recorded laughie.
4. Track the user's progress according to their prescription.

## 7. FutureWork:

### 7.1 Fixing video sharing issue

One of the features of the app is that it allows the user to share their recorded laughie. This feature uses the [share\\_plus](#) library. This feature works perfectly if the recorded laughie is audio only. However, if it is video form, the app faces an issue while reading the location of the video. This issue can be simulated by building the app, recording a video and trying to share the video from the session page.