

# **An automated big data based image retrieval system for expert shoe preference and customization**

**N.K SINGH**

Department of computer science ,BIT Mesra, nksingh27@gmail.com

**ANURAG SINHA**

anuragsinha257@gmail.com

Department of computer science and IT, UG SCHOLAR Amity University Jharkhand, Ranchi, Jharkhand (India)

## **1. Abstract**

As we all know that mobile commerce is exploding at very higher rate and everything is available just on our phones within few clicks and taps. m-commerce enables users to access online shopping platforms without needing to use a desktop computer. Examples of m-commerce include in-app purchasing are such as Amazon, Flipkart, Myntra and many more. While beautiful and mesmerizing, fashion e-commerce is a brutally competitive online business. Standing out means knowing what your customers want and using several tools of the trade of both fashion and online commerce business.

Our Application mainly focuses on footwear, providing a selection of shoes which can then be personalized by the customer. The customer can choose which area of the shoe to personalize, changing the colour and fabric to what they desire. This customization allows for a large amount of different designs to be created. And, if the customer does not want to have to design the shoes themselves, they can choose from a range of pre-made designs or buy designs other customers have made.

## **2. Introduction**

### **What is an Android?**

Android is an open source operating system for smartphone, watches, tablet TV's and computers or in other words it is an operating system for mobile devices. Android offers a very focused and unified approach for developing applications for mobile devices which means developers need to create applications only for Android, and their applications will be able to run on almost all the different android devices without any extra effort. Android was created by the Open Handset Alliance (OHA), supported by Google, and few other companies. Android is mainly based on touch gesture's i.e. direct manipulation of data on a screen that mainly related or inspired by real-world actions.

### **Android Version:**

- Android 1.0, 1.1(Apple pie)
- Android 1.5(Cupcake)
- Android 1.6(Donut)
- Android 2.0(Eclair)

- Android 2.2.x(Froyo)
- Android 2.3 - 2.3.2(Gingerbread)
- Android 3.0.x, 3.1.x, 3.2(Honeycomb)
- Android 4.0 - 4.0.2, 4.0.3 - 4.0.4((Ice\_Cream\_Sandwich)
- Android 4.1 - 4.1.1, 4.2 - 4.2.2,4.3(Jellybean)
- Android 4.4, 4.4W (KitKat)
- Android 5.0, 5.1(Lollipop)
- Android 6.0 (Marshmallow)
- Android 7.0 (Nougat)
- Android 8.0 (Oreo)
  
- Android 9.0 (Android P)
  
- Android 10 (Android Q)

### **3. Objective**

As the competition increases and quality becomes ubiquitous, customer's expectations rise. Shoppers expect to be mesmerized but sometime they face issue while shopping like oversized and unfit shoes, So can't we make an app which can measure you correctly on the phone using front camera and evaluate your body dimension by using image processing technique and suggest accordingly you perfect size of shoes which can be tailored as per the customer demand. tall, short, big and small. That's why your shoes should be made for you. This App will create custom shoes by measuring you or entering the length manually as per your requirements. Put your phone down and our machine learning algorithms measure different points for shoes. It lets you customize your entire collections by picking your favorite fabric, material, sole, thread color and more.

### **4. Modules**

The system after careful analysis has been identified to be presented with the following modules:

- **Administrator**
- **User**
- **Authentication**
- **Image Processing**

#### **Administrator:**

This module will deal with the creation of new products or services which can be added by the administrator and records of all orders which are placed with all detailed information of their personal information and physical information such as body dimension which will be

processed by algorithm in machine learning and the choice of fabric, soles, design and product type and all payment information any other analytical tools to analyze the customer and responding their queries or problem while using our service.

**User:**

This module will deal with the product representation firstly guiding them step by step that how they can avail the such service and then they can choose what they are looking for and accordingly select their choice of product and other customization such as fabric, leather, design and type of product then then can get measured by following steps guided by the app by enabling permission to their front camera. And after that they can successfully checkout with their choice of payment mode.

**Authentication:**

This module contains all the information about the authenticated user. User without his username and password can't enter into the login if he is only the authenticated user then he can enter to his login. They can also login using their social media account of Google and Facebook and directly authenticate themselves.

**Image Processing:**

This module will deal with the creation of certain algorithm in Machine Learning which will process the image of user body, height and evaluate the body dimension at the backend of the application. Which then after need to be trained with data set firstly to achieve such good model and so designs accordingly to their body dimensions.

**5. Specification:****Hardware Requirement:**

- PIV 2.8 GHz Quad Core Processor and Above
- 3 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
- 2 GB of available disk space minimum, 4 GB recommended  
(500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

**Software Requirement:**

- Android Studio 3.6
- Flutter and Dart Plugins
- Flutter SDK
- Dart SDK
- Firebase SDK

- Adobe XD
- Python 3.7
  - Open CV package

## **Implementation:**

### **What is Framework?**

A framework, or software framework, is a platform for developing software applications. It provides a foundation on which software developers can build programs for a specific platform. For example, a framework may include predefined classes and functions that can be used to process input, manage hardware devices, and interact with system software. This streamlines the development process since programmers don't need to reinvent the wheel each time, they develop a new application.

A framework is similar to an application programming interface (API), though technically a framework includes an API. As the name suggests, a framework serves as a foundation for programming, while an API provides access to the elements supported by the framework. A framework may also include code libraries, a compiler, and other programs used in the software development process.

Since they are often built, tested, and optimized by several experienced software engineers and programmers, software frameworks are versatile, robust, and efficient.

Using a software framework to develop applications lets you focus on the high-level functionality of the application. This is because any low-level functionality is taken care of by the framework itself.

### **Why do we use Frameworks?**

Developing software is a complex process. It necessitates a plethora of tasks, including coding, designing, and testing. For only the coding part, programmers had to take care of the syntax, declarations, garbage collection, statements, exceptions, and more.

Software frameworks make life easier for developers by allowing them to take control of the entire software development process, or most of it, from a single platform.

Advantages of using a software framework are:

- Assists in establishing better programming practices and fitting use of design patterns.
- Code is more secure.
- Duplicate and redundant code can be avoided.
- Helps consistent developing code with fewer bugs.
- Makes it easier to work on sophisticated technologies.
- One could create their software framework or contribute to open-source frameworks. Hence, there is a continuous improvement in the functionality.

- Several code segments and functionalities are pre-built and pre-tested. This makes applications more reliable.
- Testing and debugging the code is a lot easier and can be done even by developers who do not own the code.
- The time required to develop an application is reduced significantly.

### **Types of Frameworks**

As a developer, you should be on the lookout for frameworks that best suit your needs. Whether it is working on a website, data science, database management, or mobile applications, software frameworks exist for all genres of software programming.

There are many types of software frameworks to make it easier for developing applications for a wide range of application development domains. Let us dive into some of the software frameworks that are in vogue today:

#### **1. Android Application Framework:**

- **Flutter**
- **React**
- **Unity**
- **Xamarin**
- **Ionic**
- **Cordova**

#### **2. Web Development Framework:**

- **Flutter**
- **Angular**
- **React**
- **Laravel**
- **Ruby on Rails**
- **Django**

#### **3. Machine Learning Framework:**

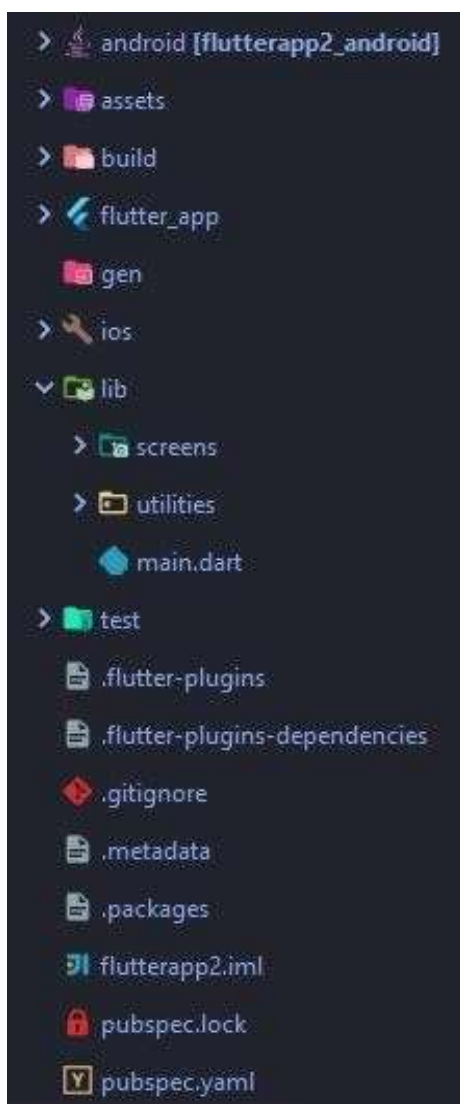
- **TensorFlow**
- **Azure ML**
- **Apache Singa**
- **Amazon Machine Learning**

- Caffe
- Spark ML lib

### What goes in a Framework?

When you install a software framework, the first thing that you need to take care of is the system requirements. Once a framework is installed and configured, it creates a directory structure.

For example, fig.(i) illustrates the directory structure of the Flutter Framework. Each of these folders could have additional directories. Directories can further have files, classes, test routines, templates, and more.



## **Programming language vs. Frameworks**

A programming language tells the computer what it should do. Every programming language features a syntax and a particular set of rules, which need to be followed every time the code is written.

A software framework is built on top of a programming language. For example, Rails, also known as Ruby on Rails, is a web framework built on top of the Ruby programming language.

Django and Flask are two different web frameworks built on top of the Python programming language. Hence, they are also known as Flutter framework which is built on top of the Dart programming language.

### **Flutter Framework**

Flutter is an open-source mobile SDK developer can use to build native-looking Android and iOS applications from the same code base. Flutter has been around since 2015 when Google introduced it and remained in the beta stage before its official launch in December 2018. Since then, the buzz around Flutter has been growing stronger.

Flutter is now the top 11 software repos based on GitHub stars. Moreover, we've already seen thousands of Flutter apps being published on app stores. One of the most notable examples is the Xianyu app created by Alibaba team, used by over 50 million people.

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Windows, Mac, Linux, Google Fuchsia and the web.

The first version of Flutter was known as codename "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit, with the stated intent of being able to render consistently at 120 frames per second. During the keynote of Google Developer Days in Shanghai, Google announced Flutter Release Preview 2 which is the last big release before Flutter 1.0. On December 4, 2018, Flutter 1.0 was released at the Flutter Live event, denoting the first "stable" version of the Framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.

### **Everything is Widget in Flutter**

The central idea behind Flutter is the use of widgets. It's by combining different widgets that developers can build the entire UI. Each of these widgets defines a structural element (like a button or menu), a stylistic element (a font or color scheme), a layout aspect (like padding), and many others.

Note that Flutter doesn't use OEM widgets, but provides developers with its own ready-made widgets that look native to Android or iOS apps (following Material Design or Cupertino). Naturally, developers can create their own widgets as well.

Flutter also provides developers with reactive-style views. To avoid performance issues deriving from using a compiled programming language to serve as the JavaScript bridge,

Flutter uses Dart. It compiles Dart ahead of time (AOT) into the native code for multiple platforms.

That way, Flutter can easily communicate with the platform without needing a JavaScript bridge that involves a context switch between the JavaScript realm and the native realm. As you can imagine, compiling to native code also boosts the app startup time.

Today, Flutter is the only mobile SDK that offers reactive views without the need for a JavaScript bridge. That's why so many mobile developers have been trying it out in their projects.

Here are some more benefits Flutter brings to mobile software development.

### Types of Widget in Flutter:

As you know in Flutter all the UI components are known as widgets. The widget which contains the code for a single screen of the app can be just of two types —

1. Stateless Widget
2. Stateful Widget

#### 6.2.1.1 Stateless Widget

Stateless widgets do not require mutable state, i.e., it is immutable.

In simple words, Stateless widgets cannot change their state during the runtime of the app, which means the widgets cannot be redrawn while the app is in action.

The structure of a Stateless widget looks like this:

A screenshot of an IDE window titled 'start\_screen.dart — demo\_app'. The code is as follows:

```
lib > start_screen.dart > StartScreen > build
1  import 'package:flutter/material.dart';
2
3  class StartScreen extends StatelessWidget {
4    @override
5    Widget build(BuildContext context) {
6      return Container(
7        // Here you can design the UI of this screen
8      );
9    }
10 }
```

The IDE interface includes a search icon, a settings gear, and a status bar at the bottom showing 'Flutter: 1.5.4-hotfix.2 No Devices'.

So, let's understand what is there in this small code snippet.

The name of this Stateless Widget is “StartScreen”, inside which we have to override the



“build” method. This build method takes in a “BuildContext” as the parameter and returns a widget. That’s why you can see that the return type of the build method is a widget. And this is the place where you can design the UI of this screen, which is Stateless.

In Stateless widget, The “build” method can be called only ONCE while the app is in action, which is responsible for drawing the widgets on to the device screen.

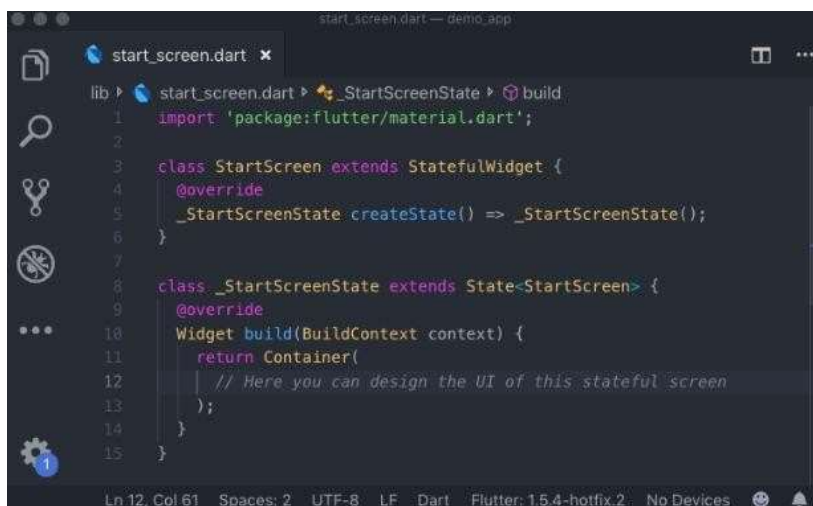
If you want to redraw the Stateless Widget, then you will need to create a new instance of the widget.

### 6.2.2.1 Stateful Widget

Stateful widgets have a mutable state, i.e., they are mutable and can be drawn multiple times within its lifetime.

They are the widgets which can change their state multiple times and can be redrawn on to the screen any number of times while the app is in action.

The structure of a Stateful widget looks like this:



```
start_screen.dart — demo_app
start_screen.dart x
lib > start_screen.dart > _StartScreenState > build
1 import 'package:flutter/material.dart';
2
3 class StartScreen extends StatefulWidget {
4   @override
5   _StartScreenState createState() => _StartScreenState();
6 }
7
8 class _StartScreenState extends State<StartScreen> {
9   @override
10  Widget build(BuildContext context) {
11    return Container(
12      // Here you can design the UI of this stateful screen
13    );
14  }
15 }
```

The name of the widget is again “StartScreen”, but now it overrides the “createState” method, instead of the “build” method, which returns the instance of the class “\_StartScreenState”.

The class “\_StartScreenState” extends from State which takes “StartScreen” as a template input. Now, this “\_StartScreenState” overrides the “build” method and returns a widget. This is where you can define the UI of the app, which is Stateful. As it is a Stateful widget you can call the build method any number of times, which will redraw the widgets on the screen.

#### So, how can we call the build method?

It’s really easy, you can use the “setState” method to call the build method, which will, in turn, redraw the widgets. This is the most important method you will need to use with any Stateful widget, to really use the statefulness of the widget.

## **Benefits of Flutter**

### **It saves you time and money**

Flutter is a cross-platform development tool. That means software developers can use the same code base for building an iOS and Android app. Cross-platform development is the best method for saving time and resources throughout the development process.

### **Excellent performance**

Flutter offers outstanding performance for two reasons. First, it uses Dart, which compiles into native code. Second, Flutter has its own widgets, so there's no need to access OEM ones. As a result, there's less communication between the app and the platform. These two features of Flutter ensure fast app startup times and fewer performance issues in general.

### **Quick development thanks to hot reload**

Flutter is gaining a lot of traction among mobile developers because of hot reload. Hot reload allows to instantly viewing the changes applied to the code on emulators, simulators, and hardware. The changed code is reloaded in less than a second. All the while, the app is running and developers don't need to waste time on restarting it.

That makes building UIs, adding new features, and fixing bugs easier. If an app encounters an error, it's usually possible to fix it and then continue using the app as if it never happened. Even if you're forced to do a full app reload, you can be sure that it's completed in no time, accelerating the development process.

### **Compatibility**

Another advantage of Flutter is the fact that it comes with its own widgets that result in fewer compatibility issues. Developers will see fewer problems on different OS versions and can spend less time on testing the app on older OS versions. Also, you can be confident that your app will work on future OS versions.

Note: Once a new version of Android or iOS comes out, Flutter widgets will have to be updated (since the tool doesn't use the native platform widgets). You might be wondering about how long it will take for Flutter to update its widgets.

Since Google is a massive internal user of Flutter, the Flutter team is strongly motivated to keep their widget sets as current and close to the platform widgets as possible. Also, Flutter widgets are customizable and can be updated by anyone. Your app can even use new widgets on older OS versions!

### **Open-source**

Flutter is an open-source technology surrounded by an active community of developers who provide support, contribute to the tool's extensive documentation, and develop helpful resources. Both Dart and Flutter are free to use.

### **The takeaway**

Flutter is one of the most innovative mobile technologies on the market right now. The advantages it brings to development teams make it a promising candidate for the mobile technology of choice in the near future.

Are you looking for a team of Flutter experts? Look no further! Our Flutter specialists know how to make the most out of this technology for a mobile development project. Get in touch with our consultants to learn how Flutter could take your project to the next level.

### **Firestore**

Firestore is a mobile and web application development platform developed by Firestore, Inc. in 2011, then acquired by Google in 2014. As of March 2020, the Firestore platform has 19 products, which are used by more than 1.5 million apps.

It is a Backend-as-a-Service — BaaS — that started as a YC11 startup and grew up into a next-generation app-development platform on Google Cloud Platform.

Firestore frees developers to focus crafting fantastic user experiences. You don't need to manage servers. You don't need to write APIs. Firestore is your server, your API and your data store, all written so generically that you can modify it to suit most needs. Yeah, you'll occasionally need to use other bits of the Google Cloud for your advanced applications. Firestore can't be everything to everybody. But it gets pretty close.

### **Functionality of Firestore:**

- Analytics — understand your users, and how they use your app
- Predictions — apply machine learning to analytics to predict user behavior
- Cloud Messaging — send messages and notifications to users
- Remote Config. — customize your app without deploying a new version; monitor the changes
- A/B Testing — run marketing and usability experiments to see what works best
- Dynamic Links — enable native app conversions, user sharing, and marketing campaigns
- App Indexing — re-engage users with Google Search integration
- In-App Messaging — engage your active users with targeted messages

### **Pros**

- Email & password, Google, Facebook, and Github authentication
- Real-time data
- Ready-made API
- Built in security at the data node level
- File storage backed by Google Cloud Storage
- Static file hosting
- Treat data as streams to build highly scalable applications

- Don't worry about your infrastructure!

#### Cons

- Limited query abilities due to Firebase's data stream model
- Traditional relational data models are not applicable to NoSQL; therefore, your SQL chops will not transfer
- No on-premise installation

#### Improve your app — stability and performance

The “improve” group of products are these:

**Test Lab** — scalable and automated app testing on cloud-hosted devices

**Crashlytics** — get clear, actionable insight into your app's crashes

**Performance Monitoring** — gain insight into your app's performance issues

#### 4. Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

#### Open CV

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Python is a general purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability.

OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

**NumPy**

NumPy is a low-level library written in C (and Fortran) for high level mathematical functions. NumPy cleverly overcomes the problem of running slower algorithms on Python by using multidimensional arrays and functions that operate on arrays. Any algorithm can then be expressed as a function on arrays, allowing the algorithms to be run quickly.

NumPy is part of the SciPy project, and is released as a separate library so people who only need the basic requirements can use it without installing the rest of SciPy.

NumPy is compatible with Python versions 2.4 through 2.7.2 and 3.1+.

**SciPy**

SciPy is a library that uses NumPy for more mathematical functions. SciPy uses NumPy arrays as the basic data structure, and comes with modules for various commonly used tasks in scientific programming, including linear algebra, integration (calculus), ordinary differential equation solving, and signal processing.

**Matplotlib**

Matplotlib is a flexible plotting library for creating interactive 2D and 3D plots that can also be saved as manuscript-quality figures. The API in many ways reflects that of MATLAB, easing transition of MATLAB users to Python. Many examples, along with the source code to recreate them, are available in the matplotlib gallery.

**Pandas**

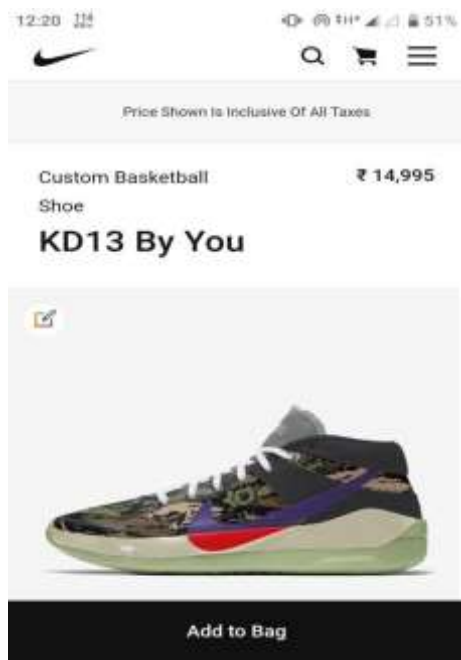
Pandas is a data manipulation library based on NumPy which provides many useful functions for accessing, indexing, merging, and grouping data easily. The main data structure (Data Frame) is close to what could be found in the R statistical package; that is, heterogeneous data tables with name indexing, time series operations, and auto-alignment of data.

**OUTPUTS**

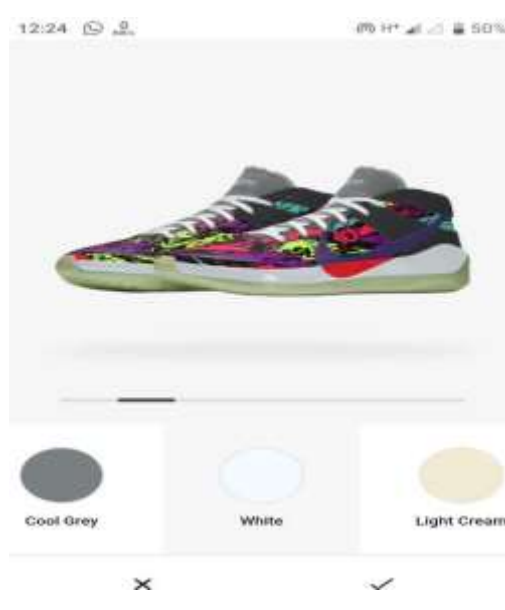
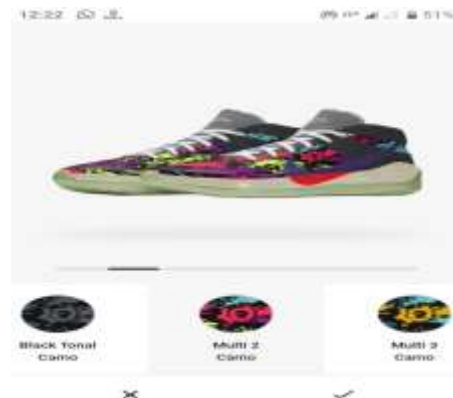
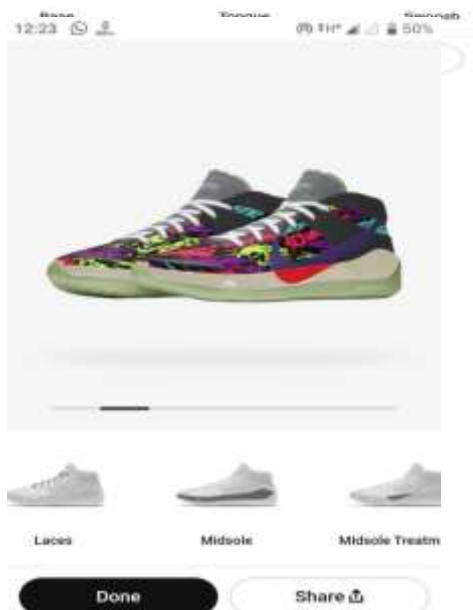
**Login details:**

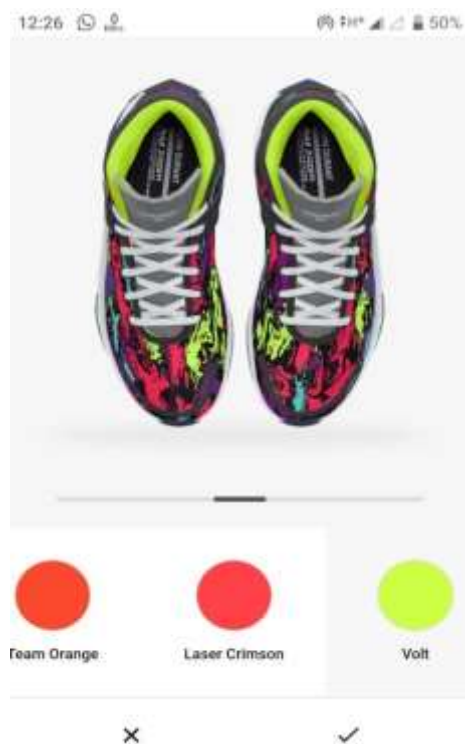
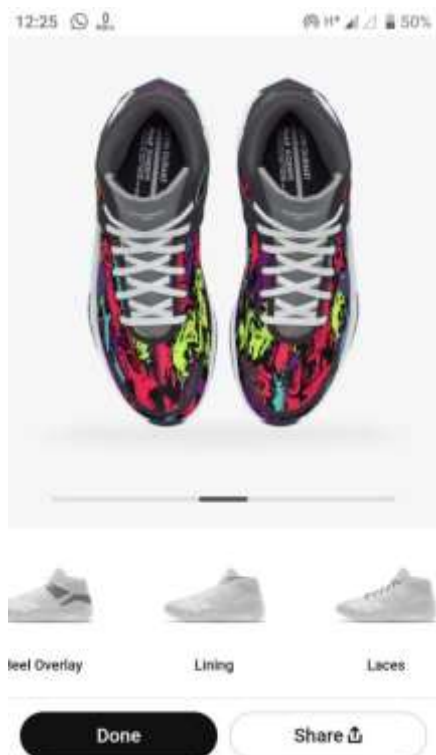
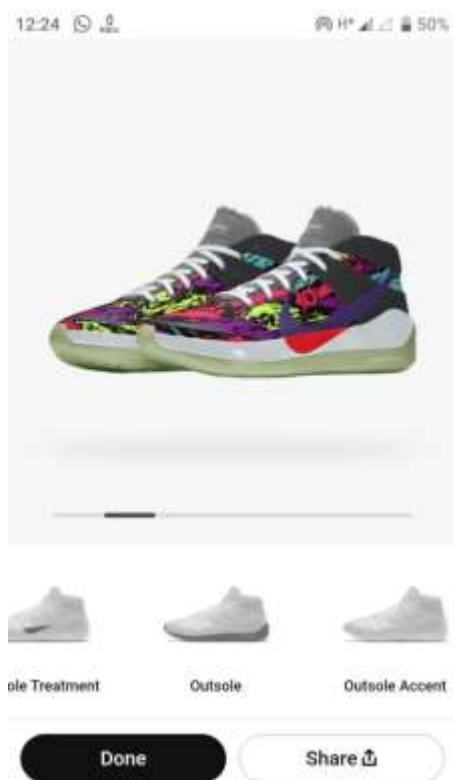


Base Model:

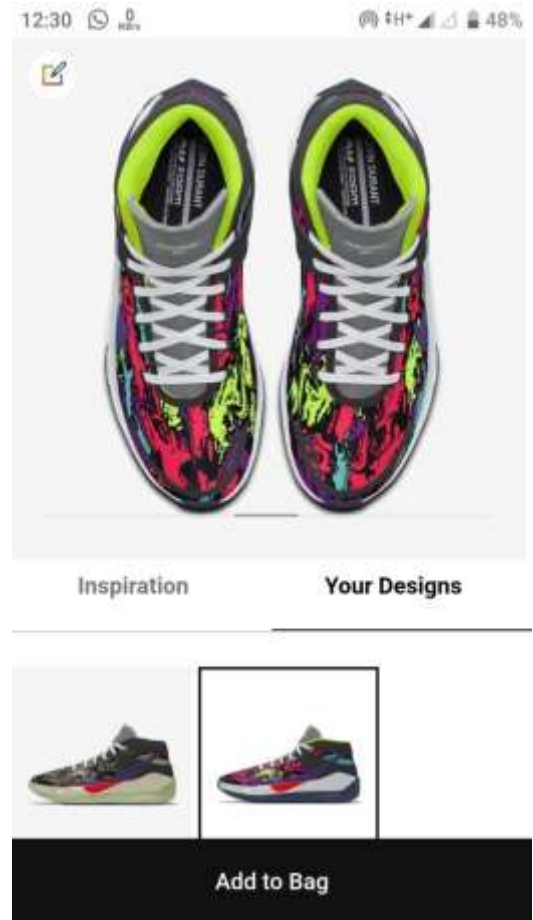
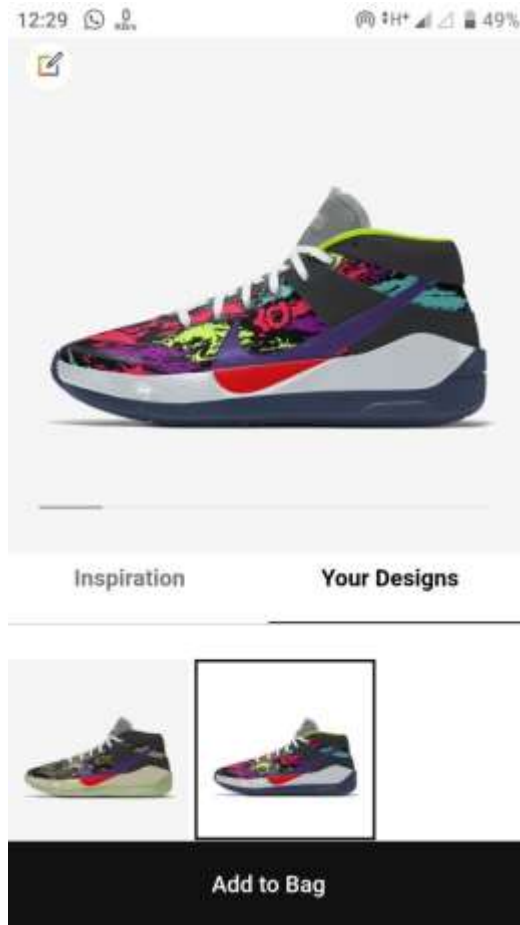


Customizing the Shoes:





**Getting the final product:**



**8. Source Codes**

**Main.dart File**

**Pubspec.yaml File**

```
name: flutterapp2
description: A new Flutter application.
```

```
# The following defines the version and build number for your
application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in
flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number
used as versionCode.
```



```
# Read more about Android versioning at
https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while
build-number used as CFBundleVersion.
# Read more about iOS versioning at
#
https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
version: 1.0.0+1
```

```
environment:
  sdk: ">=2.1.0 <3.0.0"
```

```
dependencies:
  flutter:
    sdk: flutter
```

```
# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
```

```
cupertino_icons: ^0.1.2
splashscreen: ^1.2.0
firebase_auth: ^0.16.0
dev_dependencies:
  flutter_test:
    sdk: flutter
```

```
# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec
```

```
import 'package:flutter/material.dart';
import
'package:splashscreen/splashscreen.dart';
import
'package:flutterapp2/screens/login_screen.da
rt';

void main() {
runA
pp(n
ew
Mate
rial
App(
debu
gSho
wChe
cked
Mode
Bann
er:
```

*# The following section is specific to Flutter.*

flutter:

*# The following line ensures that the Material Icons font is  
# included with your application, so that you can use the icons in  
# the material Icons class.*

uses-material-design: true

*# To add assets to your application, add an assets section, like  
this:*

*# assets:*

*# - images/a\_dot\_burr.jpeg*

*# - images/a\_dot\_ham.jpeg*

assets:

*- assets/images/icon.jpg*

*- assets/logos/facebook.jpg*

*- assets/logos/google.jpg*

*# An image asset can refer to one or more resolution-specific  
"variants", see*

*# <https://flutter.dev/assets-and-images/#resolution-aware>.*

*# For details regarding adding assets from package dependencies,  
see*

*# <https://flutter.dev/assets-and-images/#from-packages>*

fonts:

*- family: OpenSans*

fonts:

*- asset: assets/fonts/OpenSans.ttf*

*# To add custom fonts to your application, add a fonts section here,  
# in this "flutter" section. Each entry in this list should have a  
# "family" key with the font family name, and a "fonts" key with a  
# list giving the asset and other descriptors for the font. For  
# example:*

*# fonts:*

*# - family: Schyler*

*# fonts:*

*# - asset: fonts/Schyler-Regular.ttf*

*# - asset: fonts/Schyler-Italic.ttf*

*# style: italic*

*# - family: Trajan Pro*

*# fonts:*

```
# - asset: fonts/TrajanPro.ttf  
# - asset: fonts/TrajanPro_Bold.ttf  
# weight: 700
```

## 9. Conclusion

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in Dart. It also provides knowledge about the latest technology used in developing Android Application Development.

The era of mobile technology opens the windows to the android app. The websites are vanishing and the smart phones are emerging. It is time to change from conventional websites to apps which has become the part of our daily routine.

So, during this project I learned all the above things. Before this project, I had no idea about Dart, Flutter Framework and Android for making application. Although I had little bit knowledge of Framework. But now I learned a lot about Flutter Framework and got knowledge of using Dart and Firebase for developing mobile application for Android.

This is my ongoing project at lots of featured need to be developed whole algorithm and model need to made and trained using data set as currently there is no data set regarding this project is available online and can't reach out tailoring shops for gathering data set in offline mode due to spread of COVID-19 whole country is in lockdown.

So, In future I am looking forward to complete this project as It need lot of time to be developed with expertise in every field and also going to make it Open Source on Git Hub very soon as all other developers can commit to my code, Thank you !

## 10. Future Scope

It's future scope is that it can be created as a new emerging platform in fashion e-commerce as development of such application can enhance the customer satisfaction level which is required by every e-commerce brands and It's Image Processing algorithm can separately be a field of research as to come up with such good model which can predict human body dimension.

Or, it can be used as a feature in every fashion e-commerce application to guide each customer with their best fit size as everyone have not same choice or body dimensions so they can get their custom-shoes at their doorstep.

## 11. References

1. <https://flutter.dev/docs>
2. <https://firebase.google.com/docs>
3. <https://dart.dev/guides>
4. <https://developer.android.com/docs>
5. <https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/learn/>

6. <https://letsxd.com/prototyping>
  7. [https://docs.opencv.org/master/d0/de3/tutorial\\_py\\_intro.html](https://docs.opencv.org/master/d0/de3/tutorial_py_intro.html)
  8. <https://hackr.io/blog/what-is-frameworks>
  9. <https://medium.com/flutterpub/setting-intro-slider-screen-in-fastest-way-adc8c6c145c4>
  10. <https://docs.python-guide.org/scenarios/scientific/>
  11. <https://matplotlib.org/>
  12. <https://www.mdpi.com/2076-3417/9/6/1140/pdf>
  13. <https://www.sciencedirect.com/science/article/pii/S1045926X17302835>
  14. <https://www.quora.com/How-do-I-get-human-body-measurement-using-machine-learning-and-image-processing>
  15. <https://stackoverflow.com/questions/51636803/why-am-i-getting-the-stack-overflow-error>
  16. <https://www.freecodecamp.org/news/how-to-learn-flutter-in-2020/>
  17. <https://medium.com/flutter-community/how-to-learn-flutter-886c0bad8a46>
  18. <https://www.educative.io/courses/learn-dart-first-step-to-flutter>
  19. <https://www.tutorialspoint.com/firebase/index.htm>
  20. <https://developer.android.com/studio/write/firebase>
  21. <https://www.geeksforgeeks.org/adding-firebase-to-android-app/>
  22. [https://en.wikipedia.org/wiki/Flutter\\_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software))
  23. [https://www.wikiwand.com/en/Flutter\\_\(software\)#/Foundation\\_library](https://www.wikiwand.com/en/Flutter_(software)#/Foundation_library)
-