Mobile App for children at the risk of Dysgraphia: A Learning Disability

PREPARED BY

HARSHIT SHARMA(2018BTECHCSE124)

FACULTY GUIDE

Dr SONAL JAIN



Department of Computer Sciences & Engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur

May 2022

Mobile App for children at the risk of Dysgraphia: A Learning Disability

Practice School-2

Submitted in partial fulfilment of the requirements

for the degree of

Bachelor of Technology in Computer Sciences & Engineering

By:
HARSHIT SHARMA (2018BTECHCSE124)

Faculty Guide:
Dr. SONAL JAIN



Department of Computer Sciences & Engineering
Institute of Engineering and Technology (IET)
JK Lakshmipat University Jaipur

May 2022

CERTIFICATE

This is to certify that the Practice School-2 project work entitled "**DYSGRAPHIA DIAGNOSIS APP**" submitted by **Harshit Sharma (2018BTECHCSE124)**, towards the partial fulfillment of the requirements for the degree of **Bachelor of Technology in Computer Sciences & Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for Practice School-II examination.

Dr. Sonal Jain
Professor,
Department of Computer Science Engineering
Institute of Engineering & Technology (IET)
JK Lakshmipat University Jaipur

Dr. Sonal Jain
Professor,
Department of Computer Science Engineering
Institute of Engineering & Technology (IET)
JK Lakshmipat University Jaipur

Date of Submission: 18th May,2022

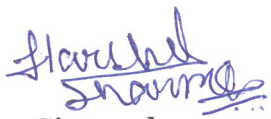
Prof. Dr. – Sanjay Goel
Director
Institute of Engineering & Technology
JK Lakshmipat University Jaipur

ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to **Prof Basant Agarwal**, Professor, Department of Computer Science & Engineering at IIIT Kota and **Prof Sonal Jain**, HoD of Computer Science Engineering JK Lakshmipat University for his valuable time and guidance that made the project work a success. We would also like to thank **Mrs Aishwarya Upadhyay** for their valuable input and Guidance on the project work.

Thanking all the friends and all those who have helped us carry out this work directly or indirectly without whom the completion of this project work was not possible.

We would also like to thank sincerely **Prof Dr. Sanjay Goel**, Director, Institute of Engineering and Technology, JK Lakshmipat University for giving us a platform to carry out the project and internships during our Practice School-2.



Sincerely yours,

Harshit Sharma
(2018BTECHCSE124)

ABSTRACT

To understand the problem we are trying to work on, in this research project. We first need to understand What dysgraphia is?, Whom does it affect?, What are its symptoms and causes?, Is there any treatment? And What we are trying to do itosolve this problem? .

Approximately 10% of the children are affected by this disorder.Dysgraphia is a neurological disorder which is characterized by writing disability. This disorder specifically affects a person's writing ability to be distorted or incorrect. In children, the symptoms of this disorder generally emerges when they are first introduced to writing. And cases of dysgraphia in adults usually occurs after some trauma or shock.The symptoms of this disorder are sized and spaced letters, or writing wrong or misspelt words, despite thorough instruction. In addition to poor handwriting, it is characterized by wrong or odd spelling, and the production of words that are not correct(i.e., using "boy" for "child").

Children with this disorder might have other learning disabilities too, but they usually have no social or other academic problems. The cause of this disorder is unknown but in adults, it usually is associated with the damage to the parietal lobe of the brain. The treatment for this disorder includes treatment of the motor disorders to help control the writing movements. Some physicians recommend that individuals with dysgraphia use computers to avoid the problems of handwriting. But this is not a permanent solution.

Dysgraphia can't be treated by medicines or vaccines, but what we are trying to do is to prevent this disorder to persist in any individual by detecting it. If we can detect this disorder in its early stages, then the individual can take measures for the disability to not last long.

Here, in this research project, we will understand how we are working to build an app and its machine-learning model to detect this disorder. We aim to focus on Early and Automated Diagnosis of Dysgraphia.

Contents

CERTIFICATE III

ACKNOWLEDGEMENTS IV

ABSTRACT V

Chapter Number	Title	Page Number
1.	***Introduction*** 1.1 Problem Statement 1.2 Research Objective 1.3 Methodology Adopted	1-2
2.	***Methodology*** **2.1 Objective** 2.1.1 Early and Automated Diagnosis of Dysgraphia 2.1.2 Build an app 2.1.3 Work on data of Dysgraphic and Non-Dysgraphic handwriting for machine learning model.	3-12
3.	***Literature Review*** **3.1Acquisition of handwriting in children with and without dysgraphia: A computational approach**	13
4.	***System Requirements Specification Document*** **4.1 Introduction** 4.1.1 Purpose 4.1.2 Document Convention 4.1.3 Intended Audience and Reading Suggestions 4.1.4 Product Scope 4.1.6 References **4.2 Overall Description** 4.2.1 Product Perspective 4.2.2 Product Function 4.2.3 User Classes and Characteristics 4.2.4 Operating Environment 4.2.5 Design and Implementation Constraints 4.2.6 User Documentation 4.2.7 Assumptions and Dependencies 4.3 External Requirements 4.3.1 User Interface	14-21

	4.3.2 Hardware Interface 4.3.3 Software Interface and Requirements 4.3.4 Communication Interface **4.4 System Features** 4.4.1 API Calls **4.5 Other Non-Functional Requirements** 4.5.1 Performance Requirements 4.5.2 Safety Requirements 4.5.3 Security Requirements 4.5.4 Software Quality and Attributes 4.5.5 Business Rules **4.6 Other Requirements**	
5.	***Software Design Document*** **5.1 Introduction** 5.1.1 Purpose 5.1.2 Scope 5.1.3 Overview 5.1.4 Definitions and Acronyms **5.2 System Overview** 5.3 System Architecture 5.3.1 Architectural Design 5.3.2 Decomposition Description **5.4 Design Rationale** **5.5 Data Design** 5.5.1 Data Description 5.5.2 Data Dictionary **5.6 Component Design** **5.7 Human Interface** 5.7.1 Screen Images and Actions	22-33
6.	***Conclusion and Future Scope***	34
	Appendix	

CHAPTER 1 : INTRODUCTION

1.1 Problem Statement

Dysgraphia is a learning condition that manifests itself in difficulties with writing. It's a neurological condition that affects both children and adults. Dysgraphia causes people to use wrong wording for what they're trying to say, and write that are difficult to read. Approximately 10% of the children are affected by this disorder. The aetiology of dysgraphia isn't always clear, yet it can occur after a traumatic experience in adulthood. You can acquire methods to assist you to overcome some of the obstacles that the disorder poses in school and life once it has been diagnosed.

What we are trying to do is to prevent this disorder to persist in any individual by detecting it. If we can detect this disorder in its early stages, then the individual can take measures for the disability to not last long.

Here, in this report, we will understand how we are working to build an app and its machine-learning model to detect this disorder. We aim to focus on Early and Automated Diagnosis of Dysgraphia.

1.2 Research Objectives

OBJECTIVE 1: Early and Automated diagnosis of Dysgraphia

OBJECTIVE 2: Build an App as proposed in Objective 1.

OBJECTIVE 3: Work on data of Dysgraphic and Non-Dysgraphic handwriting for machine learning model.

1.3 Methodology Adopted

1) Methodology adopted for Objective 1: Early and Automated diagnosis of Dysgraphia

- Brainstorm how we can solve this problem?
- Solution proposed after brainstorming was to develop an app to somehow detect Dysgraphia at its early stage.

2) Methodology adopted for Objective 2: Build an App as proposed in the Objective 1

- Make a prototype of how the app will look using any tool such as Figma, Adobe XD, etc.
- Decide the workflow of the App.
- Identify and decide on what technologies and frameworks the App will be developed and will work on.

(a)Front-End Technology: Flutter will be used to develop the aesthetics and design of the App.

(b) Back-End Technology: In the Back-End technology, there will be only one factor working and that is for the machine learning model we will be developing a TensorFlow model using python.

3) Methodology adopted for Objective 3: Work on data of Dysgraphic and Non-Dysgraphic handwritings for machine learning model.

- Collect the handwriting Data.
- Decide what features we want to extract from the data, and test and train the data.
- Identify and determine what machine learning algorithm best suits our research project.
- Build a model that could be integrated with the App.

CHAPTER 2 : Methodology

2.1 Objectives

2.1.1 Early and Automated diagnosis of Dysgraphia

1. **Brainstorm over how we can solve this problem:** It is a problem-solving technique that involves the spontaneous contribution of ideas from all members of the group working on a particular cause.

2. **Solution of the Brainstorming:** Building an App, is the solution to the problem that came up if we can somehow detect Dysgraphia and its symptoms in a way that users can also use the app with the utmost ease. We decided we want to achieve "Early and Automated Diagnosis of Dysgraphia".

3. **Decide what all features should be present in the app:** The app should consist of several screens from which users can easily navigate:

   (i). Home Screen: The Home Screen is displaying what causes we are working on. It is responsible to display the crucial aim we are trying to achieve using this App.

   (ii). Guidelines Screen: The Guidelines Screen is responsible for making the user understand what all rules need to be followed to use the features of the App.

   (iii). Discover Screen: The Discover Screen is displaying steps and procedures the app is supposed to work in that way.

   (iv). Testing and Get Result Screen: The testing and Get result screen is responsible for giving helpful feedback after the user has uploaded a dysgraphic sample. The user first needs to select a image either from the gallery or capture an image using the camera. After the selection of the image, the result will automatically be applied to the image and it will display the result of the sample being dysgraphic or non-dysgraphic.

   (v). Blogs Screen: The Blog Screen is displaying various information and activities that have been going on related to dysgraphia.

(vi). AboutUs Screen: The About Us Screen is displaying how can user connect with the team of the app and can get additional help as well. As of now it's displaying only our email so that users can reach us.

2.1.2 Build an App as proposed in the Objective 1

1. **Make a prototype of how the app will look using any tool such as Figma, Adobe XD, etc.:** It is the most crucial part before developing any app. A developer must know what all requirements and design aesthetics are expected which need to be followed. Usage of platforms like Figma and AdobeXD is important in drawing out the wireframe of the app.

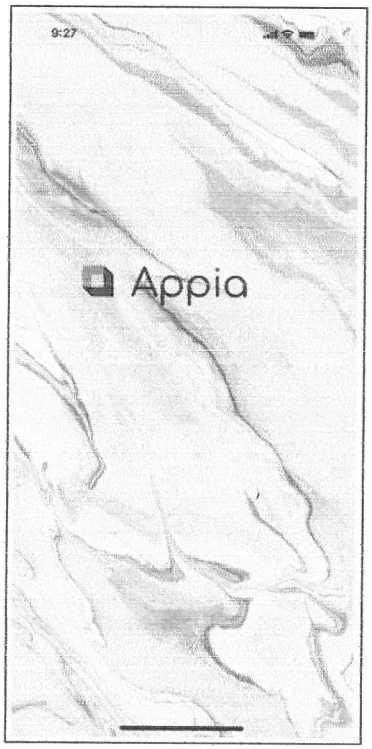


Fig. 1: Introduction Screen

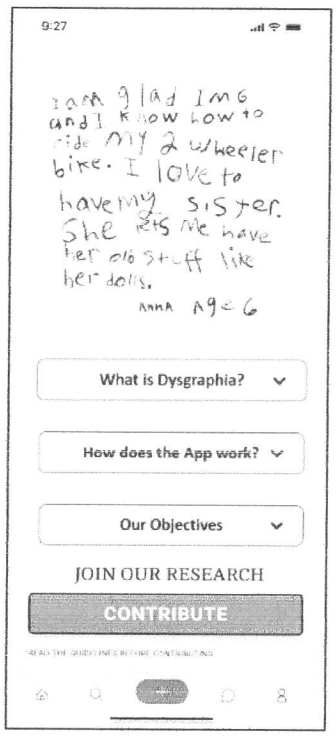


Fig. 2: Home Screen

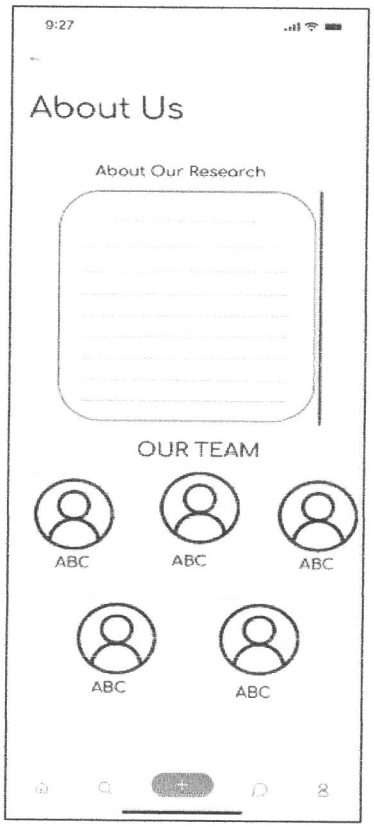


Fig. 3: AboutUs Page

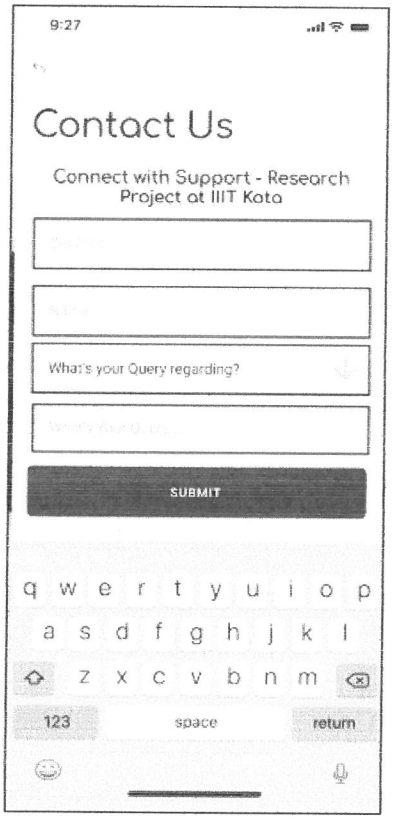


Fig. 4: Contact Us Screen

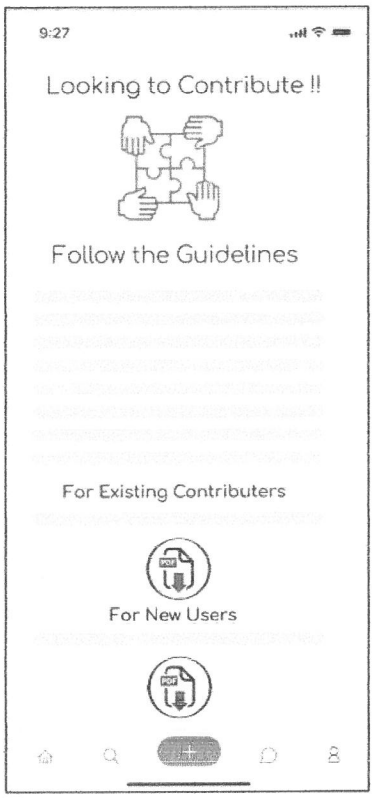


Fig. 5: Guidelines/Rules Screen

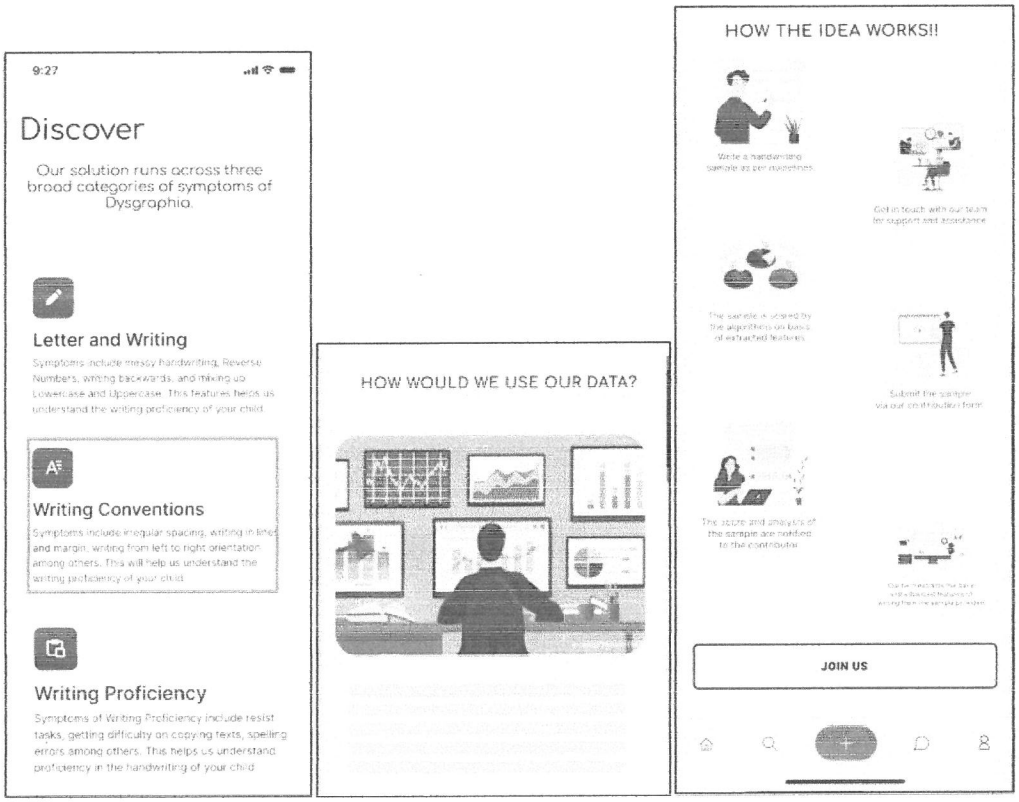


Fig. 6: Discover Screen: Where the user can find how the app works.

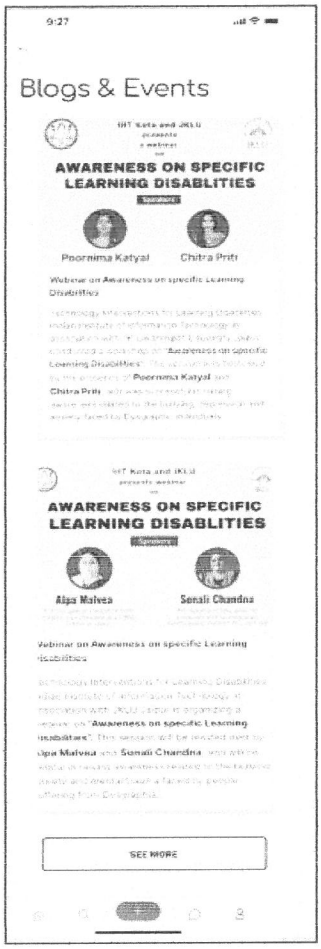


Fig. 7: Blogs and Events Screen

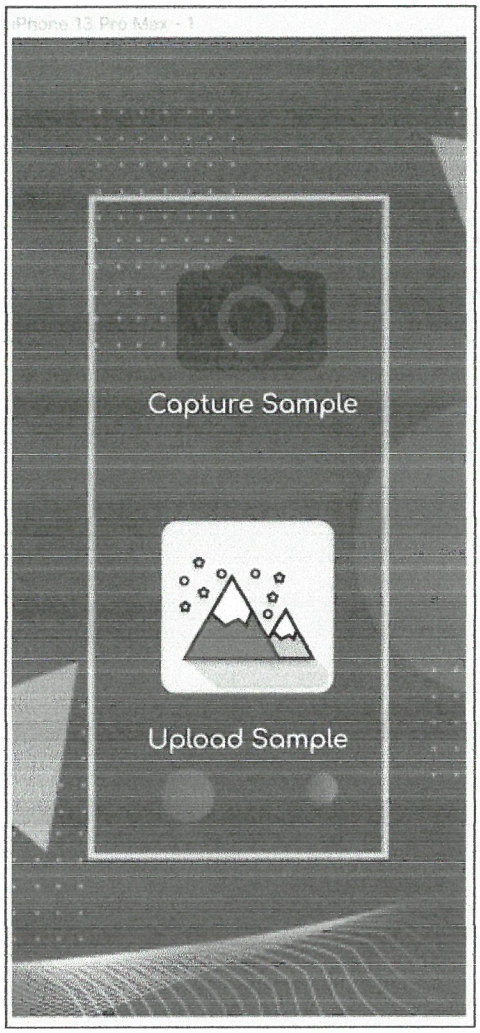


Fig. 8: Choose an option to fetch the image from

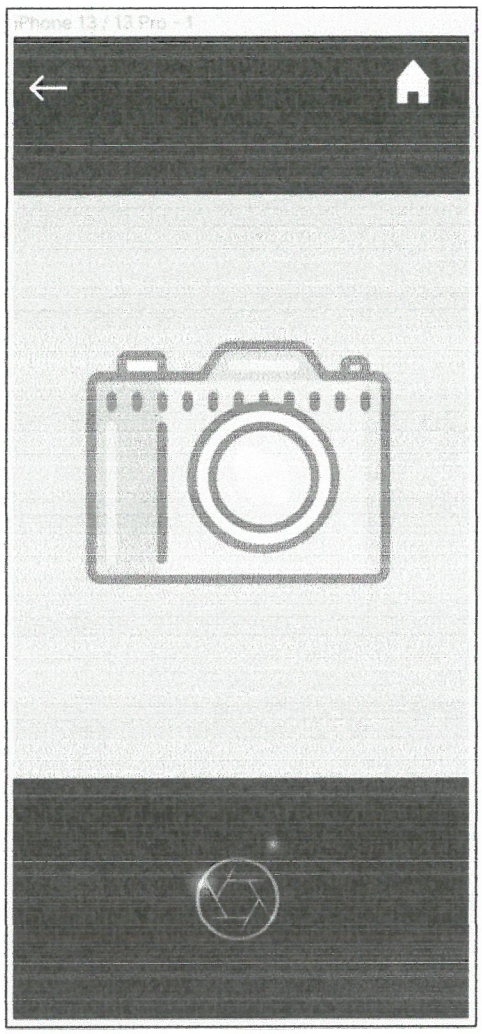


Fig. 9: Camera Screen

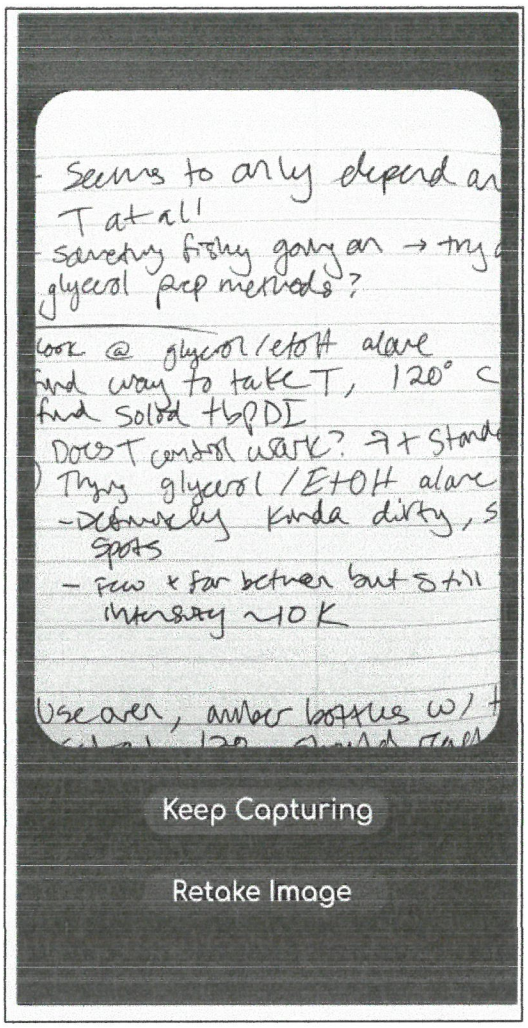


Fig. 10: Screen after Capturing Image

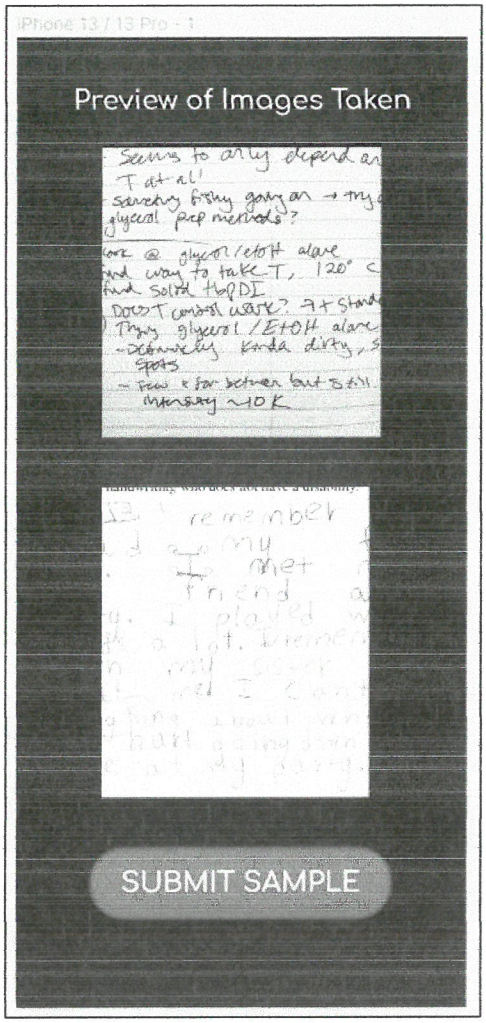


Fig. 11: Submit Captured Image Screen

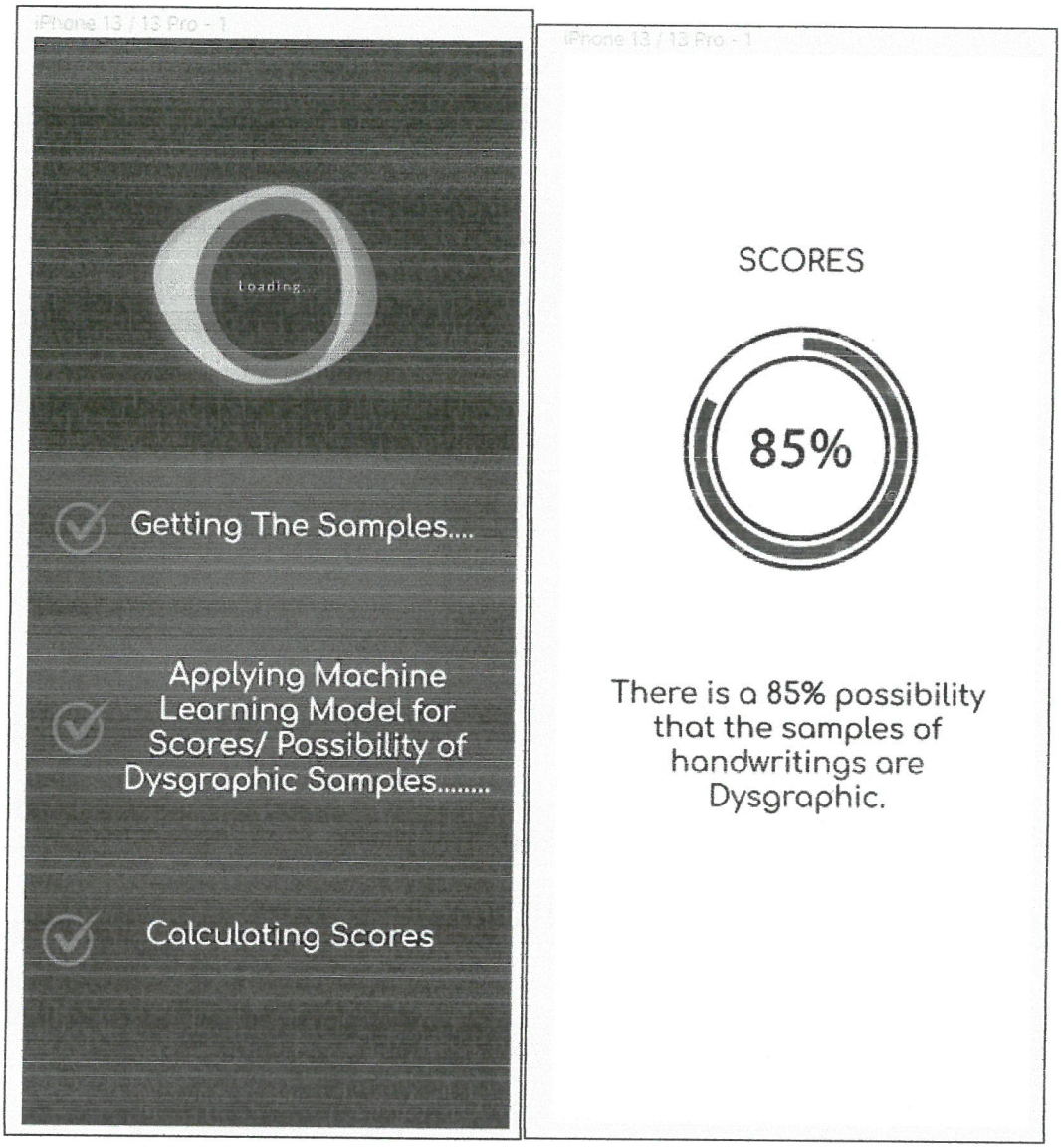


Fig. 12: Processing and Result Screen

2. **Decide the workflow of the App:** After the wireframe, we must decide on how the app's features will work, we must decide on a systematic flow.

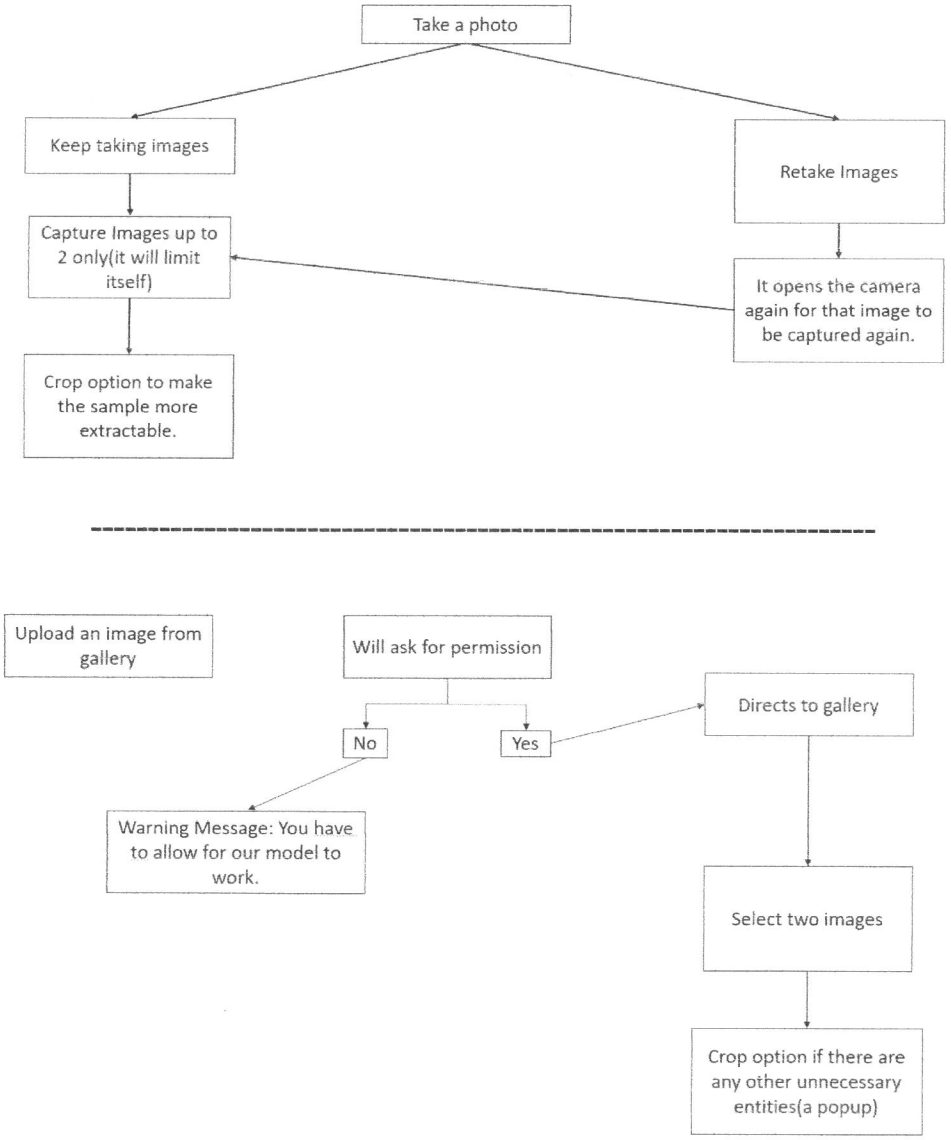


Fig. 13: How the fetching of Images works in the App

3. **Identify and decide on what technologies and frameworks the App will be developed and will work on:** We must decide on what technologies, tools and frameworks we will work on to make the development of the application a success.

i). Languages Used:

- **Flutter:** For the front-end Development.

Flutter is an open source UI software development kit by Google. It is used to develop cross-platform applications for android, iOS, Linux, macOS, Windows, etc. Major Components of Flutter Architecture include:

- Dart Platform
- Flutter Engine
- Foundation Library
- Design-specific Widgets
- Flutter Development Tools

Most of the work in Flutter uses Widgets, Widgets are of three types: Stateless Widgets, Stateful Widgets and Inherited Widgets. These three basic widgets are used in the development of every Flutter application.

Although the instances of widgets are immutable, the function of the above mentioned widgets are as follows:

- Stateful Widgets allows interaction between user and application by giving access to setState method which then maintains the state in separate state objects. In simple words If a widget can change when user interacts with it is a Stateful Widget.
- Stateless Widgets act as a constant such that before displaying anything changes the widget has to be recreated. In simple words widget that never changes are stateless like Icons, Icon button and Text are examples of Stateless Widgets.
- Inherited Widgets allows another widget to inherit its property and functioning allowing the state to be passed down to its children.

- **Python:** For the Machine-Learning Model(Tensorflow/TensorFlow Lite). Although, There is a different team working on the machine-leaning model building. But there is a slight knowledge regarding the machine learning model in section 2.1.3.

ii). IDE's Used:

- **Android Studio:** For the front-end development of the app using flutter.
- **Jupyter Notebook:** For development of the Machine-Learning Model.

iii). Tools Used: Teachable Machines to train and get powerful machine-learning models for demo.

2.1.3 Work on data of Dysgraphic and Non-Dysgraphic handwriting for machine learning model.

1. **Collect the handwriting Data:** Collecting data before developing any machine-learning model is very crucial because it's like laying strong evidence for machines to learn from. For a machine to be learning it needs to be trained with a lot of images of Dysgraphic and Non-Dysgraphic Handwriting.
2. **Decide what features we want to extract from the data, test and train the data:** It's a very important step towards the development of the machine-learning model as we will train the machine to decide whether a handwriting sample is dysgraphic or Non-Dysgraphic based on these features only.
    - Baseline
    - Pressure
    - Variance of letter size
    - Variance of line spacing
    - Slant angle
    - Variance of word spacing

If we find any abnormalities in the handwriting samples based on the features above that sample is a Dysgraphic Sample.

3. **Identify and determine what machine learning algorithm best suites our research project:** For this machine-learning model we have used the following algorithms:
    - **Random-Forest Classifier:** Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. In this app we dealing with a classification problem only as we need to classify between Dysgraphic and Non-Dysgraphic handwriting samples. We are getting an accuracy of 97%.

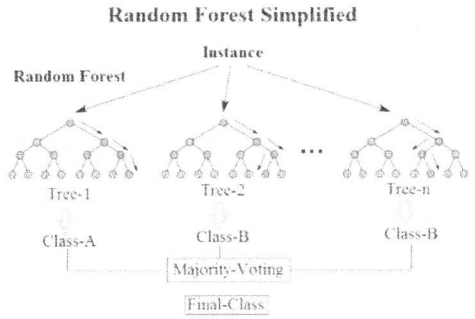


Fig. 14: Working of Random Forest

- **Support Vector Machine(SVM):** Set of supervised learning methods used for **classification, regression and outliers detection**. The advantages of support vector machines are Effective in high dimensional spaces. Still effective in cases where a number of dimensions is greater than the number of samples. We got an accuracy of 69% after using SVM.

- **Logistic Regression:** To obtain an odds ratio in the presence of more than one explanatory variable. As we working on different variables in this project we need to use this machine learning algorithm. We got an accuracy of 79% using Logistic Regression

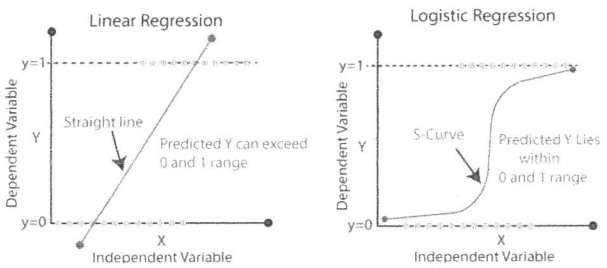
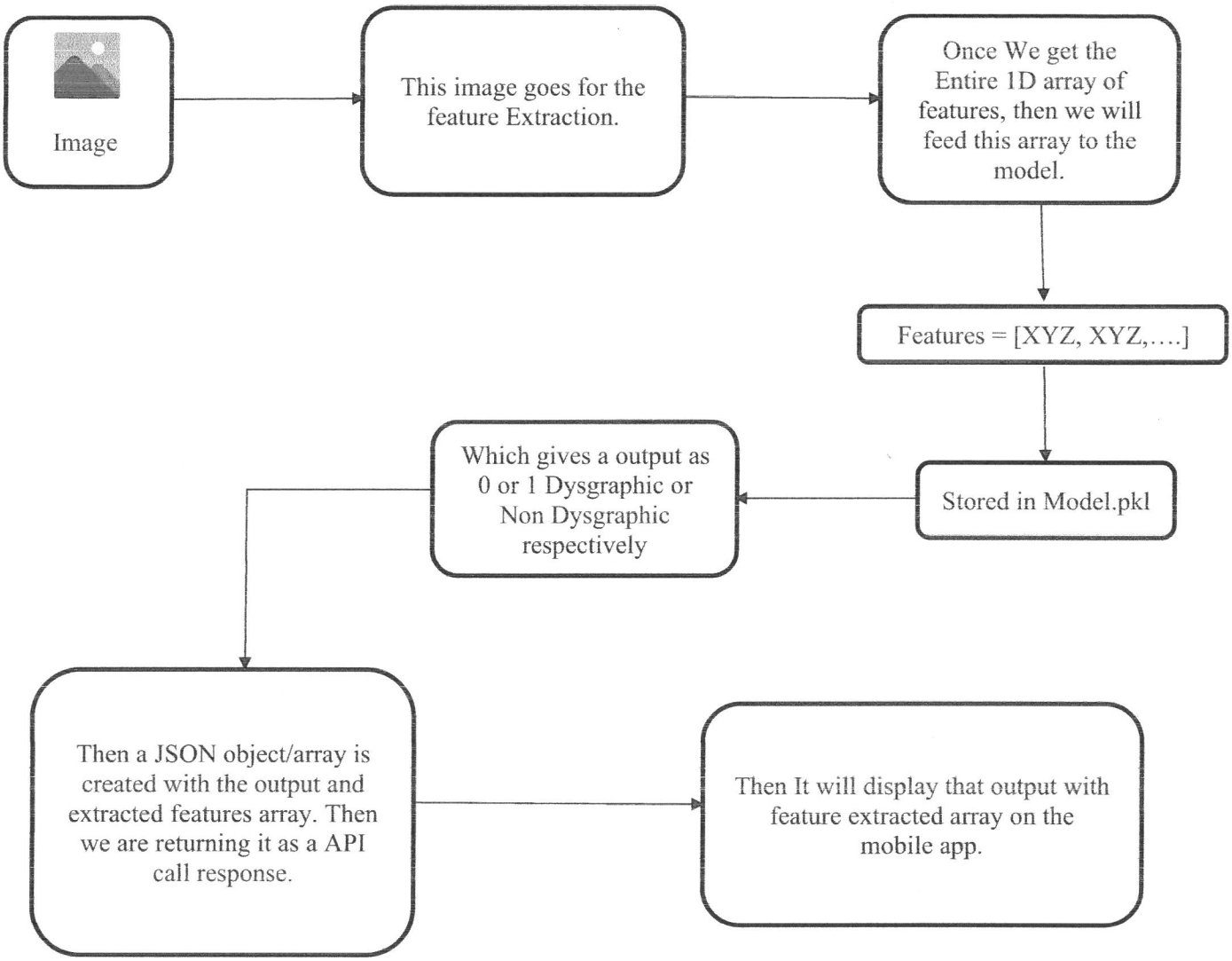


Fig. 15: Comparison between a logistic and linear regression

4. **Build a model that could be integrated with the App: How to integrate an ML model?**
    - After the completion of training and building the machine-learning model in python.
    - The firstst Step, would be to use the TensorFlow Lite converter to convert our TensorFlow model with the app.
    - The second step, would be to add the tensorflow lite model to the asset folder.
    - Third Step, would be to make changes in build.gradle file
    - Build a function that will load the model and will print out the results.

The above work done is for a demo model that is to be integrated with the app.

Actual Model Integration(Flow Diagram)

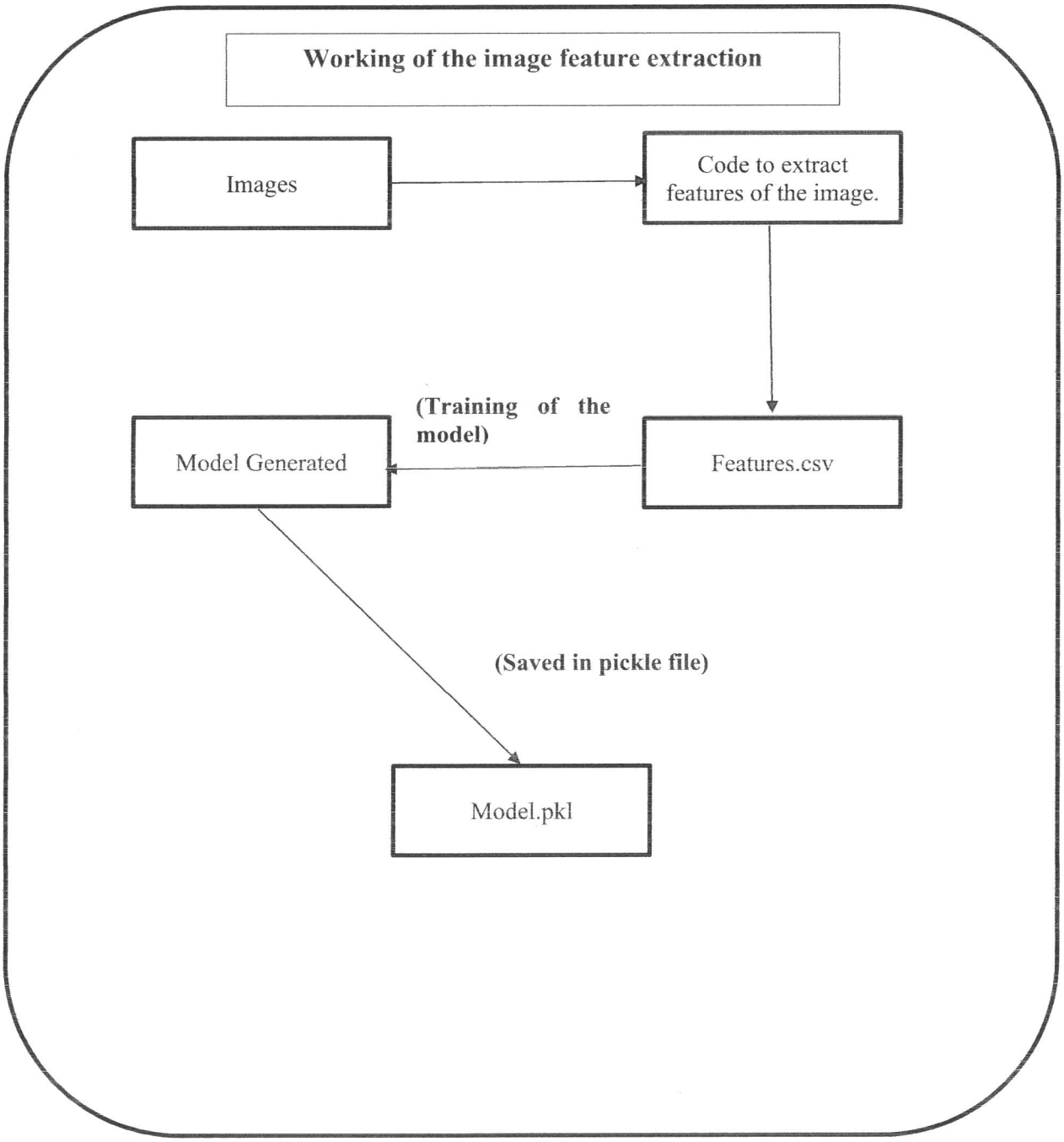


Machine-Learning Model Results displaying Method.

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. In our project we need the mobile interface to work with model.pkl file.

Files included in the API Server are:

- **Model.pkl**
- **Code to extract the features**



CHAPTER 3 : Literature Review

3.1 Acquisition of handwriting in children with and without dysgraphia: A computational approach

Handwriting is a complex skill to acquire and it requires years of training to be mastered. Children presenting dysgraphia exhibit difficulties automatizing their handwriting. This can bring anxiety and can negatively impact education. 280 children were recruited in schools and specialized clinics to perform the Concise Evaluation Scale for Children's Handwriting (BHK) on digital tablets. Within this dataset, we identified children with dysgraphia. Twelve digital features describing handwriting through different aspects (static, kinematic, pressure and tilt) were extracted and used to create linear models to investigate handwriting acquisition throughout education. K-means clustering was performed to define a new classification of dysgraphia. Linear models show that three features only (two kinematic and one static) showed a significant association to predict the change of handwriting quality in control children. Most kinematic and statics features interacted with age. Results suggest that children with dysgraphia do not simply differ from ones without dysgraphia by quantitative differences on the BHK scale but present a different development in terms of static, kinematic, pressure and tilt features. The K-means clustering yielded 3 clusters (Ci). Children in Cl presented mild dysgraphia usually not detected in schools whereas children in C2 and C3 exhibited severe dysgraphia. Notably, C2 contained individuals displaying abnormalities in term of kinematics and pressure whilst C3 regrouped children showing mainly tilt problems. The current results open new opportunities for automatic detection of children with dysgraphia in classroom. We also believe that the training of pressure and tilt may open new therapeutic opportunities through serious games.

CHAPTER 4 : Software Requiremnt Specifications Document(SRS)

4.1 Introduction

4.1.2 Purpose

The product that is being developed is Software application free to use for everyone. The app is designed to get a early and automated diagnosis of a disorder namely "Dysgraphia", the patients of this disorder usually have learning and writing disorders but a early diagnosis can be very supportive in the curing of the disorder. The product focuses on early and automated diagnosis of the disorder Dysgraphia. ***The idea of the app is to detect the disorder using the sample of writing written by the individual with the disorder. In order to detect that we will be training a machine learning model to extract the features like baseline, pressure, Average Letter size, average line spacing, slant angles, and average word spacing.***

4.1.3 Document Conventions

When writing the Software Requirement Specification document for the Sofware Application "Dysgraphia Diagnosis App"designed for a social cause the following standards are being followed to make document effective and readable:

- The whole SRS is in Justified alignment.
- The chaper Heading are in 14 font size, Bold Times New Roman Font Style.
- The theory of each topic is written in normal Times New Roman with a font size of 14.
- The main heading is labelled as 4.1, 4.2, 4.3 and so on, as SRS is chapter 4 in the report.While the subheadings are labelled as 4.1.1, 4.1.2 or 4.2.1, 4.2.2 and so on.
- The SRS consist of 1.5 line spacing after every line.

4.1.4 Intended Audience and Reading Suggestions

This document is written for:

- Researchers who are studying Disgraphia.
- Developers and programmers looking to develop an app like this.
- Project Managers
- Designers
- Testers
- Document Writers
- Users involved in the modification of the Software Application who are going to suggest changes and give feedback.

This document consists of the various steps and procedures to develop the Software Application "Dysgraphia Diagnosis App". The following sections describes product function, Scope and other overall description regarding this research project work.

4.1.5 Product Scope

The Software designed is a Machine-learning based application using flutter as a front-end technology and python as a back-end technology for the machine-learning model. For example, if an individual is suffering from Diabetes they have a remote solution to check whether their sugar is high or low, that machine helps that individual to detect the problem early so that he/she can go for a cure. In the same way we are trying to provide a diagnosis system which can be remotely accessed by everyone so that the individual suffering from Dysgraphia can go for an early cure to lead a healthy life.

The final product will enable users to save human resources and time to get a easy and automated diagnosis of the disorder. This increase the chances to cure the disorder at an early stage without any hinderance.

4.1.6 References

References that I used to create the SRS document for the client are noted below:

- SRS Template by Karl E. Wiegers
- https://www.coursehero.com/file/p355itf/12-Document-Conventions-When-writing-then-SRS-document-for-Atlantic-Lottery/
- Research Article: Acquisition of handwriting in children with and without dysgraphia: A computational approach
- Website: https://tild.co.in/index

4.2 Overall Description

4.2.2 Product Perspective

The idea of the product is to make people aware of these types disorders and provide an easy solution for the diagnosis. This diagnosis system will help people to check whether a child or a person is suffering from this disorder and if they already taking a cure then what's the improvement on their situation. The app has the capability of answering most of the queries related to Dysgraphia Disorder and is capable of diagnosis as well.

4.2.3 Product Function

The product function has various parts in it like there is a navigation bar which helps in the navigation from different screens, the screen name and their functionality are as follows:

- ***Home Screen:*** This screen will make sure that the user can understand what dysgraphia as a disorder means.
- ***Discover Screen:*** This screen is responsible to make users understand how our app is working and shows how hassle free procedure that we are providing for the diagnosis.
- ***Guidelines Screen:*** The user will understand how to proceed with the diagnosis of the patient and what rules they need to follow.
- ***Add Image and Get Result Screen:*** After understanding the rules on how to give the sample of the handwritten image, they can either choose the image from the gallery or from the camera for the image to be analysed so that they can get the result of the person being Dysgraphic or not (on the same screen). This is the part of the application where the machine-learning model is applicable.

4.2.4 User Classes and Characteristics

The two user classes for this system are:

- **API Users:** It consists of application developers who want to learn and research more about Dysgraphia as a disorder.
- **Application visitors who want to get a Diagnosis:** It consists of non-technical users who want to just use the facility of the app to get a diagnosis result.

4.2.5 Operating Environment

In the operating environment for this software, it is assumed that the user who is using this service has the following:

- Android or iOS(preferable above Android 4.4 and above & for iOS 5)
- Access or Connectivity to the internet for accessing better services apart from offline use.

4.2.6 Design and Implementation Constraints

- For developing this application flutter is responsible for the front-end design and for back-end python which was used in the machine-learning model.
- Language: The application that was developed only supports standard English.

4.2.7 User Documentation

- The user must understand the rules and guidelines on how to give handwritten samples.
- Except that there is no documentation/user manual required for the user as it is a simple software in which users just need to understand how to give their samples. For that, the app is designed in such a way that users will be familiar with how can they use the software.

4.2.8 Assumptions and Dependencies

The assumptions are as follows:

- There will be no third-party interruptions.
- There will be no security or privacy breach, it is completely safe to use for the users.
- The Software will not ask for any user information as far as this version of the application is considered.
- The Software will solve most of the queries of the users regarding this disorder.

The dependencies are as follows:

- The admin and only admin can use the information obtained by the software application to improve the functionality of the app.
- There are no such specifications for users to consume/use the Software(Dysgraphia Diagnosis App).

4.3 External Interfaces Requirements

4.3.1 User Interfaces

The User Interface should always be easier to use and navigate. Therefore, the application should let the user navigate easily to every screen. Also, the app in itself is very explanatory for the user to use.

The user interface looks like any other software these days people are more comfortable using a navigation bar, there is a navigation bar present in the application for easy use. There must be buttons present for an interactive user experience.

4.3.2 Hardware Interfaces

The Minimum Requirements for this application to work are as follows:

- 2 GB of RAM in their mobile phone.
- Internet
- Decent Storage for the app to take an image and fetch images easily. (preferable to the standard storage of devices these days)

4.3.3 Software Interfaces and Requirements

This application is developed on the following frameworks and development environments are:

- Android Studio
- Android Emulator
- Jupyter Notebook
- Tensorflow and Tensorflow Lite
- Teachable Machines

4.3.4 Communication Interfaces

The Software Application needs help while communicating/ installing the app on the device

- Play Store
- Network Server Communication Protocol etc.
- Internet Service Provider
- Broadband Internet
- It's not uncommon for mobile apps to **use HTTP** for some parts, and **HTTPS** for others. They can also **do** some other tricks which normal browsers don't generally touch.

4.4 System Features

The major services provided by the product are as listed below:

- Making sure that the user understands the cause towards which we are working.
- Clarifies the queries regarding Dysgraphia Disorder
- Collect data and improve the functioning of the Application.
- Customer Satisfaction By giving answers to their questions with utmost ease.

4.4.1 API Calls

- The user will submit an image which will be processed and analysed by the machine-learning model. (**Priority 1**)
- After the diagnosis the user will get the results, and then the client must be able to scan another sample of the image without any error. (**Priority 1**)
- The server must send all request that is sent by the app.(**Priority 1**)
- The server must reply to all the requests from the app. (**Priority 1**)
- **Stimulus/Response Sequences**
  - **Input:** The user request **Priority 1**
  - Analysing image samples using machine learning. **Priority 1**
  - **Output:** Result of the diagnosis with its confidence.

4.5 Other Non-Functional Requirements

4.5.1 Performance Requirements

The following are the performance requirement of the product that we are developing:

- The app must respond to all the requests/ activities
- It must not take a long time to process any activity or action means the application should not buffer or lag while interacting with the user.
- User interface must be easy to understand. So, that it could be used in the easiest possible way.

4.5.2 Safety Requirements

There are no safety requirements identified or specified.

4.5.3 Security Requirements

The following are the Safety requirements kept in mind before developing the application:

- The Application is only allowed to collect data from the user/client interactions only.
- It must not intervene with the user's system functioning.
- It should be safe to use which means it should not allow any cyber-attacks on the user's machine.
- It should not leak the user's privacy means the data of the user and admin must remain unpublished and classified.

4.5.4 Software Quality Attributes

Software quality attributes for the software application that is useful for both user and developers are listed below:

- Adaptability: If the user captures an image which not clear and cannot be analysed. The machine learning model must give a warning.
- Availability: The Dysgraphia Diagnosis App must be able o attend to multiple users at the same time. It must not show an error if there are too many people interacting with it at once. Although, the traffic on the app still needs to be defined.
- Correctness: The Software application developed must satisfy its users by correctly responding to each request.
- Maintainability: The software application should be easy to maintain so that it could be updated with information with utmost ease.
- Ease of Usage: The software application must be easy to use.

4.5.5 Bussiness Rules

There are no Business rules specified as of now. But if needed it implements all the standards of business policies and decisions. This includes all the rules and regulations that a user and developer should abide by. This includes cost factors and the use of illegal protocols, neither users nor admins can break these rules and regulations.

4.6 Other Requirements

The final application should be able to handle any kind of unexpected error to prevent any kind of data loss from the database.

CHAPTER 5: Software Design Document

5.1 Introduction

5.1.1 Purpose

This document describes the architecture and system design of the Software Application (Dysgraphia Diagnosis App).

5.1.2 Scope

The software application developed aims at helping people recognize disorders and make quick steps to cure them. The automated diagnosis enhances the chances of the person making improvements remotely by daily analysing the diagnosis of the cure suggested by the professionals.

- It aims at helping people with an early, daily and automated diagnosis of Dysgraphia.
- Queries related to Dysgraphia can be solved on the app itself.

5.1.3 Overview

The Software Design Document's objective is to make sure how the software development will take take place. It ensures both parties agree on the deliverables so there is no confusion. The software application is a Dysgraphia Diagnosis App which will help in the Early and Automated Diagnosis of Dysgraphia.

- It is developed using Flutter which will be involved in the front-end of the app.
- The machine learning model developed working in the back-end will be developed using python(TensorFlow).
- The app will try to provide and make people aware of the information about Dysgraphia as a disorder.

5.1.4 Definition and Acronyms

There are no Acronyms used for this Software Design Document. Also, there is no definition that needs to be defined.

5.2 System Overview

- Functionality: The software we designed has a complex procedure in the back-end as it uses machine-learning technologies but its functionality on the front end is flawless.
- Context: It was developed as an application that can be used to diagnose people with Dysgraphia a disorder in which an individual has difficulties in writing and learning.
- Design of the Project: It would be a simple to use design for users, they can navigate easily and most of the things on the apps are self-explanatory.

5.3 System Architecture

5.3.1 Architectural Design

Please refer to Chapter 2 of this document to understand how the architecture of the app took place.

Below is a block diagram of a simple explanation of how the Diagnosis will work:

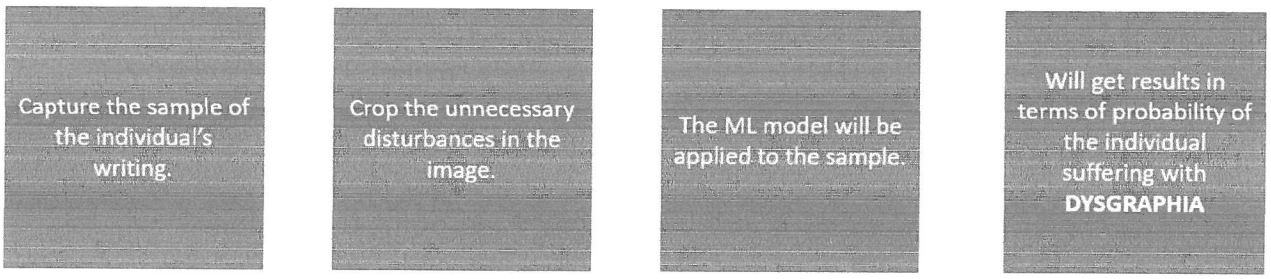


Fig. 16: Block Diagram of the apps Architecture on the Diagnosis Screen

5.3.2 Decomposition Description

- **Product Screen:** The user will simply click on the icon of the app and start using the app. He can navigate to any screen after opening the app.
- **Home Screen:** The Home Screen is displaying on what causes we are working. It is responsible to display the crucial aim we are trying to achieve using this App.
- **Guidelines Screen:** The Guidelines Screen is responsible for making the user understand what all rules need to be followed in order to use the features of the App.
- **Discover Screen:** The Discover Screen is displaying steps and procedures the app is supposed to work in that way.
- **Testing and Get Result Screen:** The testing and Get result screen is responsible for giving helpful feedback after the user has uploaded a dysgraphic sample. The user first needs to select an image either from the gallery or capture an image using the camera.

After the selection of the image, the result will automatically be applied to the image and it will display the result of the sample being dysgraphic or non-dysgraphic.

- **Blogs Screen:** The Blog Screen is displaying various information and activities that have been going on related to dysgraphia.
- **AboutUs Screen:** The About Us Screen is displaying how can user connect with the team of the app and can get additional help as well. As of now its displaying only our email so that users can reach us.
- **Admin Screen:** The admin will only have a database as an extra feature to maintain the data. Although, the admin screen is yet to be developed that is for the future goal.

5.4 Design Rationale

The journey of designing this app starts by prototyping and designing an interactive workflow using Figma but keeping a sober design. It involves rigorous help from online design ideas the app to make a prototype of the app. After that, we make a lot of changes in the design of the app only after that do we get on to the development of the Application.

I collaborated with my mentors and colleagues to complete this project. There were two teams one working on the machine-learning model and one on the front-end & integration technologies.

This is the first-ever app designed to diagnose a disorder which aims at "Early and automated Diagnosis of Dysgraphia".

This app will surely be helpful to the people out there who find it difficult to understand what is going on with their child or someone close to them with this disorder.

5.5 Data Design

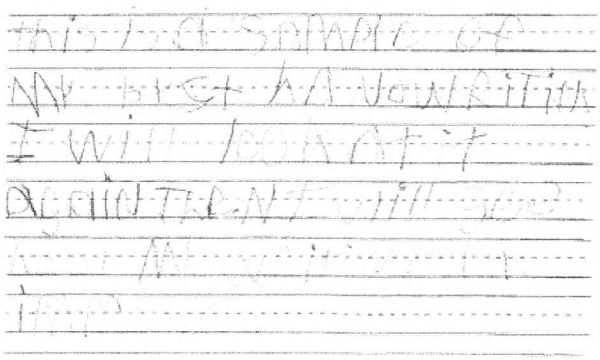
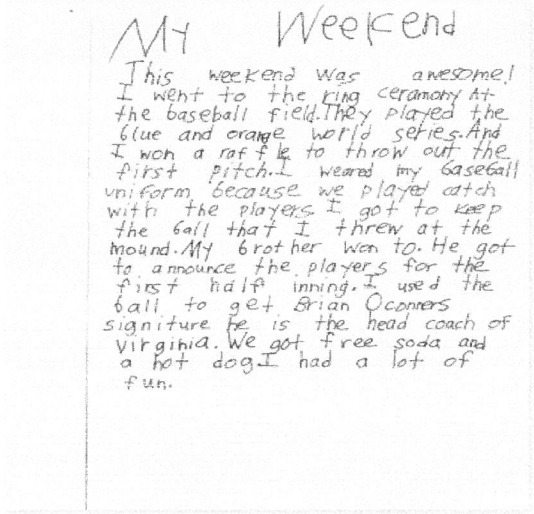
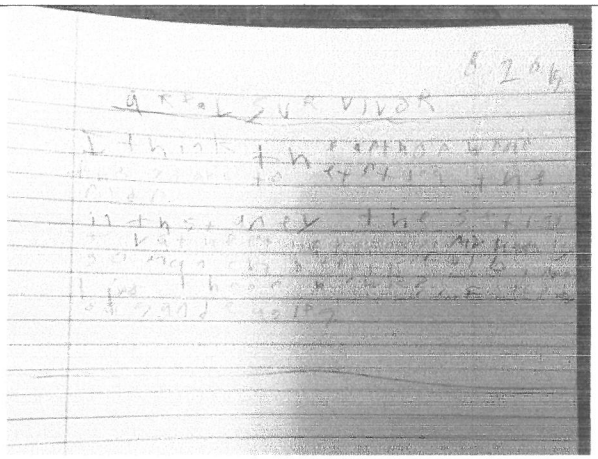
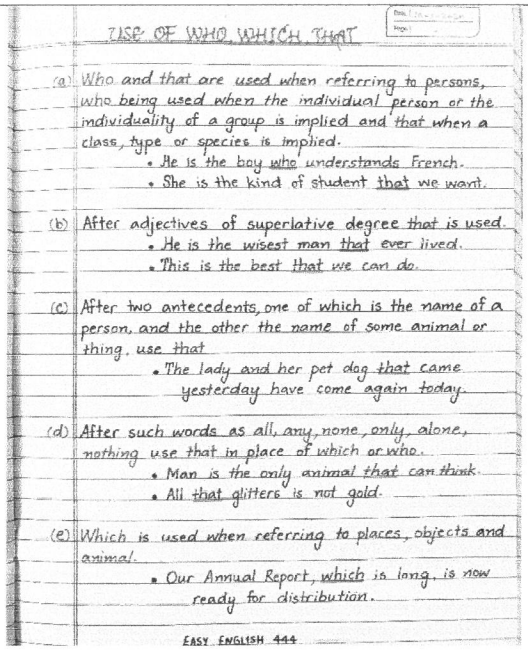
5.5.1 Data Description

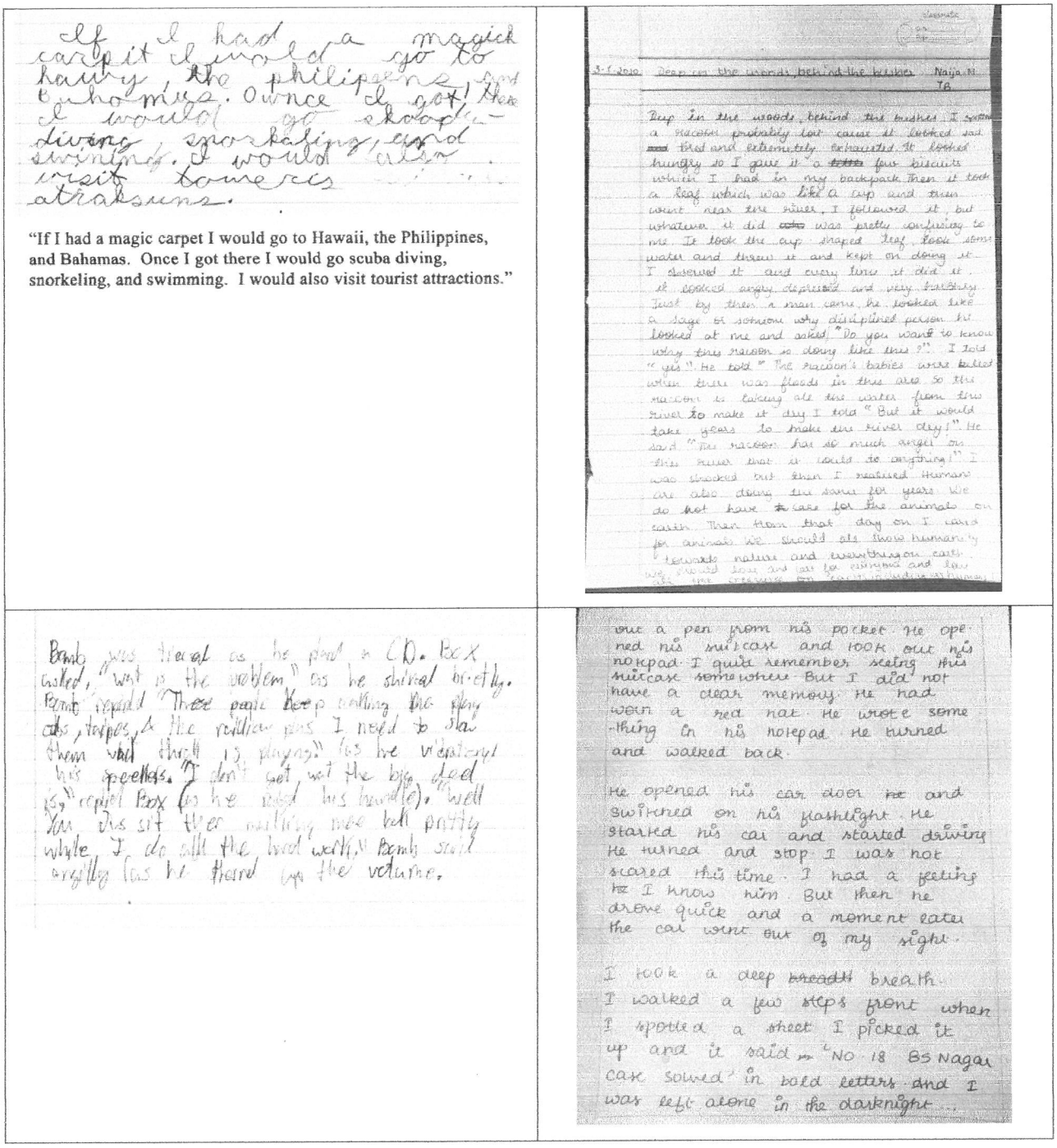
Our data consists of a different bunch of images that are Dysgraphic and Non-Dysgraphic Images.

- We have about 100 sample images of the patients suffering from Dysgraphia.
- And about 80 sample images of normal handwritten people.

5.5.2 Data Dictionary

There are lot of images in the dataset some of them are as follows:

Dysgraphia Samples Images	Non-Dysgraphic Sample Images
Figure 1 An example of a second grade student's handwriting with dysgraphia 	 My Weekend This weekend was awesome! I went to the ring ceramony At the baseball field. They played the blue and orange world series. And I won a raffle to throw out the first pitch. I weared my baseball uniform because we played catch with the players. I got to keep the ball that I threw at the mound. My brother won to. He got to announce the players for the first half inning. I used the ball to get Brian Oconners signiture. he is the head coach of virginia. We got free soda and a hot dog. I had a lot of fun.
	 USE OF WHO, WHICH, THAT (a) Who and that are used when referring to persons, who being used when the individual person or the individuality of a group is implied and that when a class, type or species is implied. • He is the boy who understands French. • She is the kind of student that we want. (b) After adjectives of superlative degree that is used. • He is the wisest man that ever lived. • This is the best that we can do. (c) After two antecedents, one of which is the name of a person, and the other the name of some animal or thing, use that • The lady and her pet dog that came yesterday have come again today. (d) After such words as all, any, none, only, alone, nothing use that in place of which or who. • Man is the only animal that can think. • All that glitters is not gold. (e) Which is used when referring to places, objects and animal. • Our Annual Report, which is long, is now ready for distribution. EASY ENGLISH 444



"If I had a magic carpet I would go to Hawaii, the Philippines, and Bahamas. Once I got there I would go scuba diving, snorkeling, and swimming. I would also visit tourist attractions."

There are more images by which we are training our machine learning model. These are just some samples with a kind of comparison table between Dysgraphic and Non-Dysgraphic.

5.6 Component Design

There are different components design that were kept in mind before developing this app are as follows:

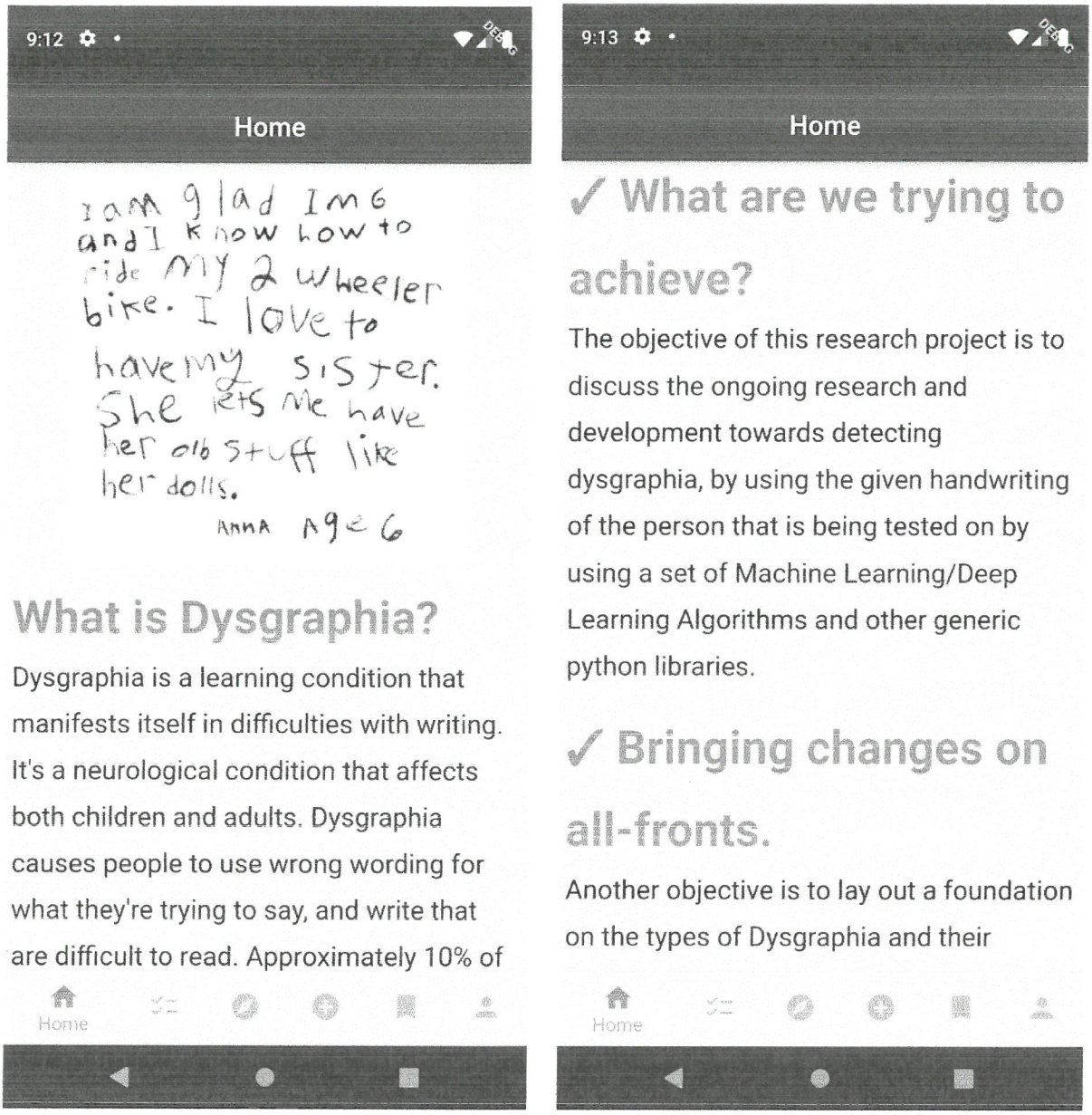
- The whole should be in light mode.
- There should be a navigation bar at the bottom.
- There should icons on the navigation bar for interactive design.
- The Font of the app should be normal(in flutter we have default font size) followed throughout the App.
- There should be a screen title on every navigation screen.
- There should be scrolling on the discover page for steps to be displayed in an interactive manner.
- The whole app should be in the Dark Blue colour Scheme.
- The unselected icons should be grey.
- The heading on any navigation Screen should be bold and in blue colour.
- The font colour on the app should be in black colour.
- Every Component should follow some intended schemes correctly so that the application looks good.

5.7 Human Interface

Please, refer to section 5.3 subsection 5.3.2. in this same chapter to understand what will be the Human Interface.

The app is designed and developed in such a way that the user experience will not be hindered or ruined.

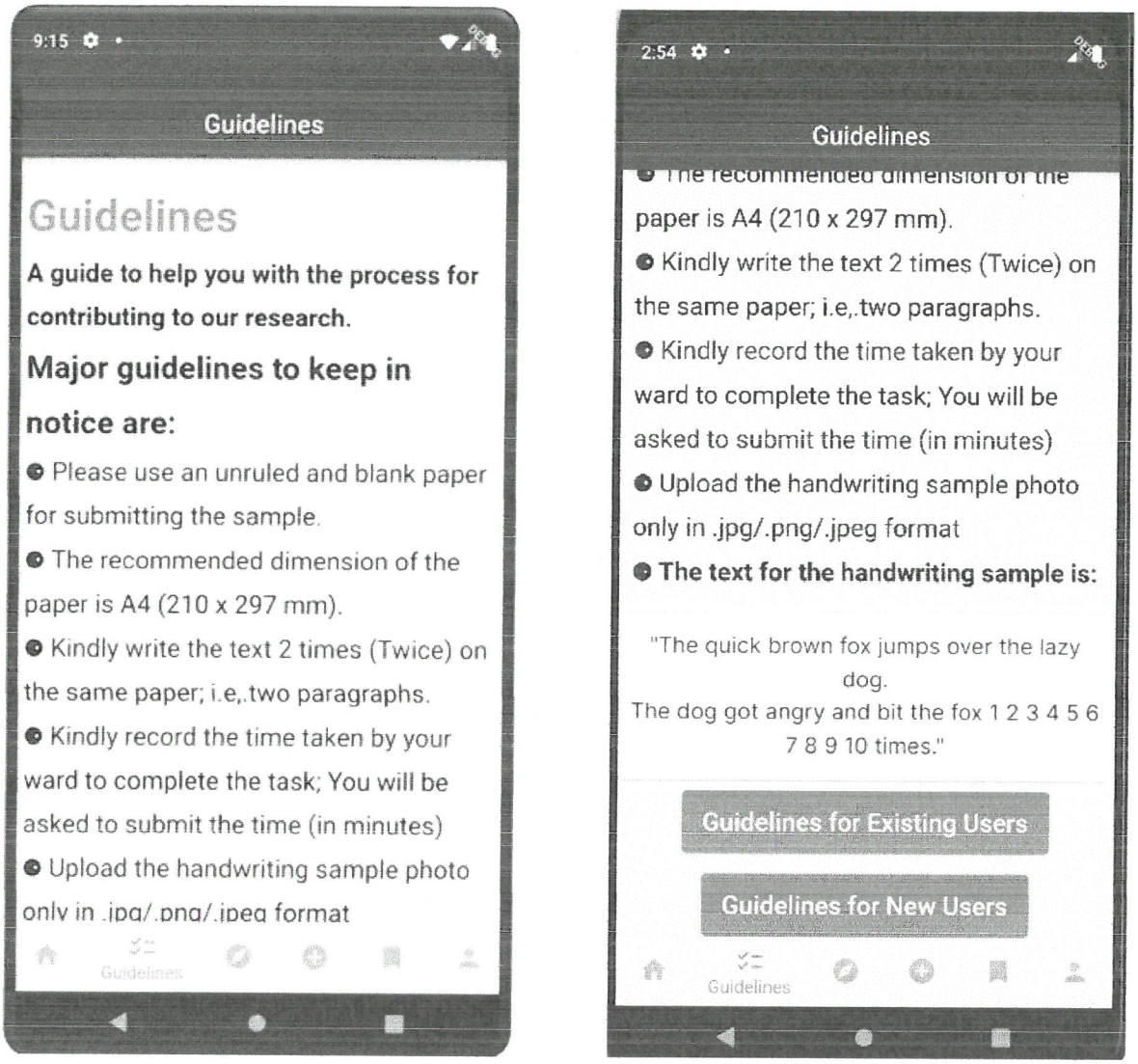
5.7.1 Screen Images and Actions



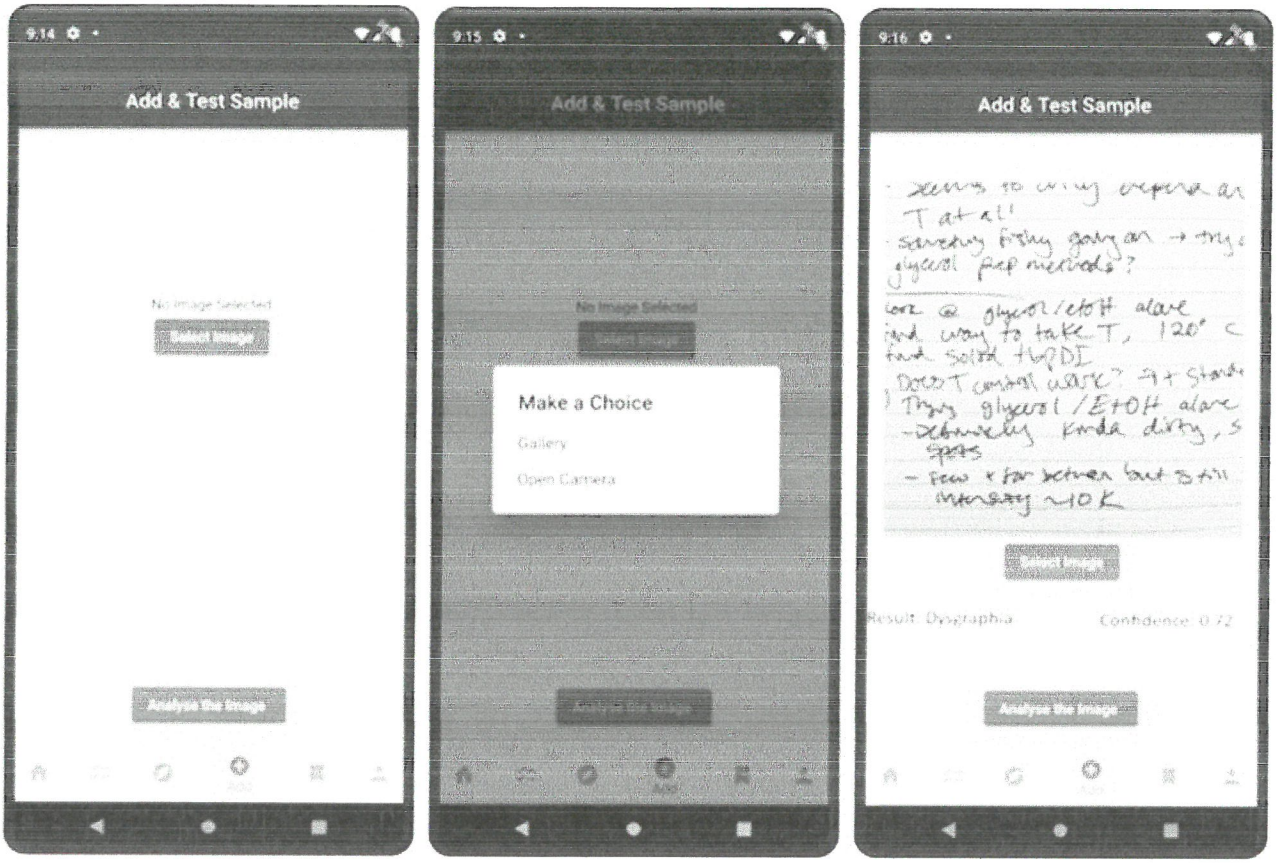
HOME SCREEN: The user will be able to know what is the issue that we are working on.



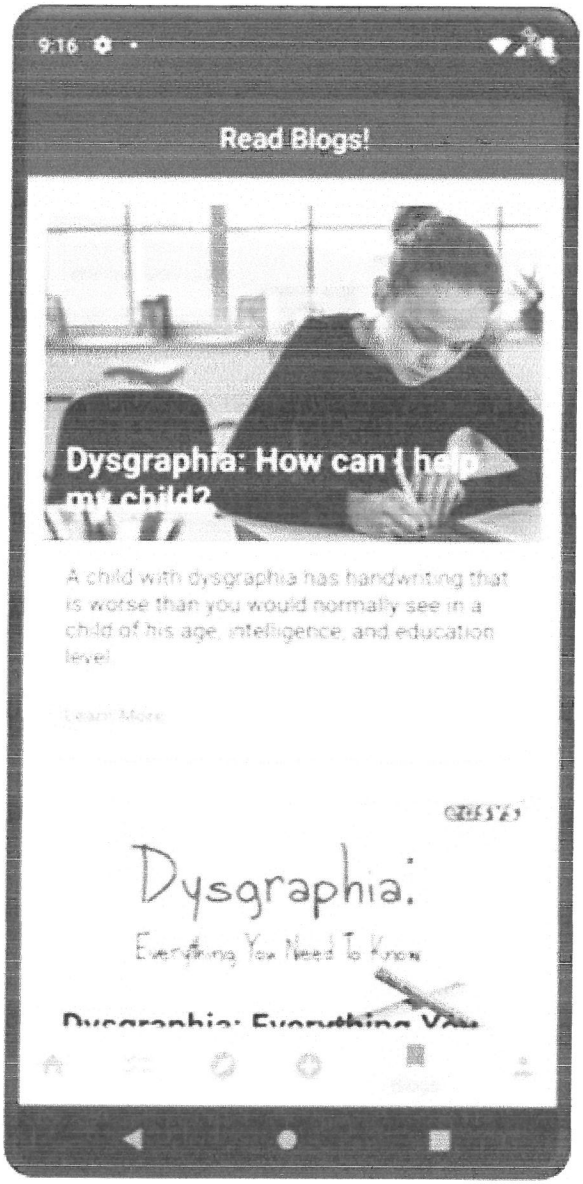
DISCOVER PAGE: User will understand the process/steps on how to achieve the results and how does the app works.



GUIDELINES PAGE: This page is helpful in understand how to capture the dysgraphia image sample.



Add & Test Sample Page:Get the result after adding



BLOGS PAGE:

User will be able to know more and get aware about this disorder via blogs and videos on this page.



ABOUT US PAGE:

Users will be able to know about the developers and will be able to contact us.

CHAPTER-6: Conclusions and Future Scope

In the conclusion, We were able to develop a Software Application Dysgraphia Diagnosis App using flutter and python(Tensorflow).

The application developed is now capable of making the user understand what causes this app works and how to use the app?. This software will not only help to diagnose the patients but will also make people aware of and recognise the disorder. These types of disorder should never be overlooked.

Future Scope:

Although the app is fully functional there are some features that will make the app more interactive and more controllable.

- There should be an admin's side to this app so that database and data can be collected for the improvement of the application.
- There should be a feature that will save all the previous results in the form of a pdf in the app itself.
- Connect an API for the blog's screen for regular and latest updates on this disorder.

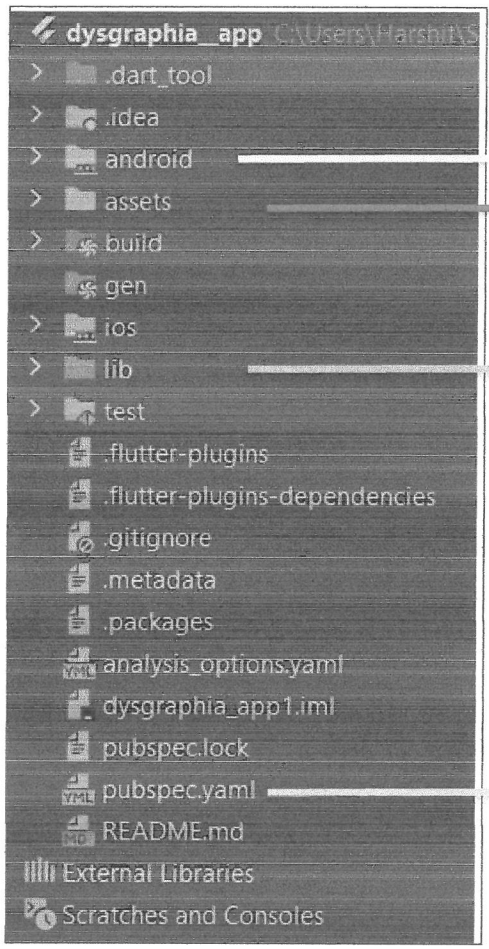
Discussion:

In all, it was a wholesome and awesome experience to understand how software in the real world is being implemented and what steps we need to follow to achieve software application functionality. We learnt how software documentation is a must if we want to compile our work, propose a solution or even if we want to make a project. Every step in the development of an App is essential from idea generation, and rough design, to layman's implementation to software requirement specification(SRS), and software development design(SDD), we have to document each and every step towards the software development of an application.

It was a very good experience for me as Every process has utmost professionalism in it, From meetings to team management to development process. And I hope the skills that I have developed while doing this project will help me in my future.

APPENDIX

The hierarchy of the file system in the development of the Application are as follows:



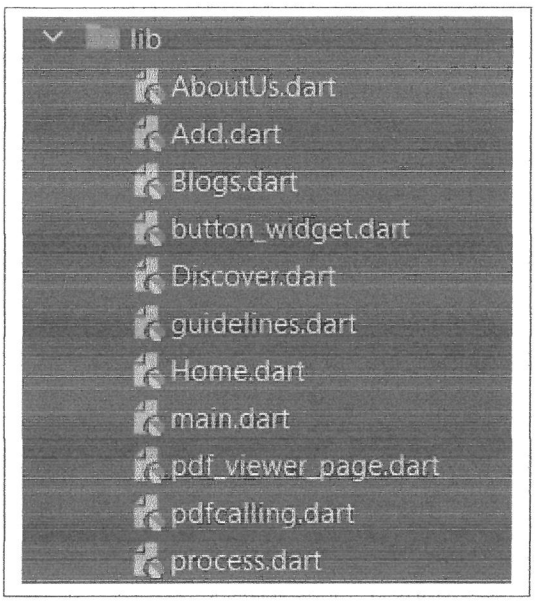
Android folder: Contains files and folders required for running the application on an Android operating system.

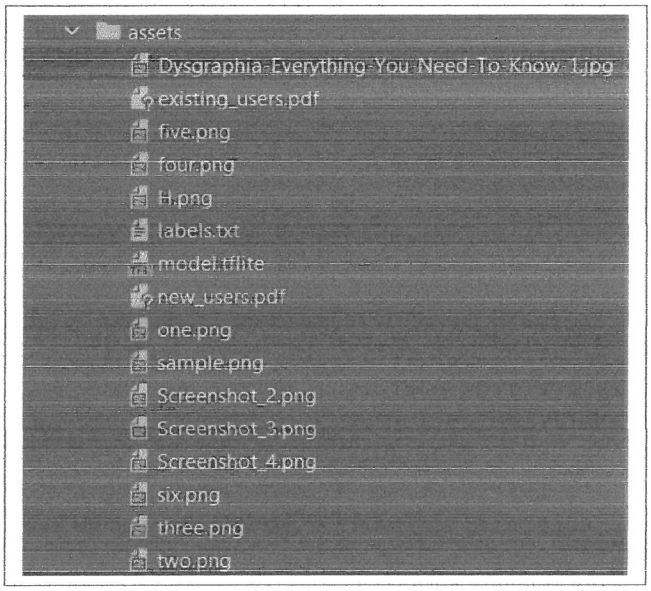
Asset Folder: Contains all the files that should be included with the app such as image files, video files or any machine learning model as a tflite extension.

Library folder: The most important folder in the project, used to write most of the dart code

Pubspec.yaml File: Located at the root of your project, to identify assets and pre-built plugins required by an app.

Files inside library folder and Asset folders are as follows:





Asset Folder:

- All the .png, .jpg extension are the images used inside the App.
- Model.tflite is the tensorflow lite model of the machine learning model.
- Labels.txt contains the category of images that we want to classify using machine learning model.
- .pdf files are the files which contain all the rules and regulations for the existing and new users of the application.

Dart Code of the Lib Folder with their Description:

Main.dart file: Dart programs has an entry point called main. When you run flutter or dart file it first runs main function. It calls all the files needed for the development of the App.

```
//this the main.dart file which runs on building the app and
calls all the files function and plugin to make an app work.
import 'package:flutter/material.dart';
import 'package:dysgraphia_app1/AboutUs.dart';
import 'package:dysgraphia_app1/Add.dart';
import 'package:dysgraphia_app1/Blogs.dart';
import 'package:dysgraphia_app1/Discover.dart';
import 'package:dysgraphia_app1/Home.dart';
import 'package:dysgraphia_app1/guidelines.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
//add light theme
  @override
  Widget build(BuildContext context) {
    return    MaterialApp(theme:    ThemeData.light(),    home:
MyBotNavBar());
  }
}

class MyBotNavBar extends StatefulWidget {
  const MyBotNavBar({Key? key}) : super(key: key);

  @override
  State<MyBotNavBar> createState() => _MyBotNavBarState();
}

class _MyBotNavBarState extends State<MyBotNavBar> {
//setting the page which will be displayed when the app opens
and when a page is selected
  int _selectedIndex = 0;
  final List<Widget> _children = [
    HomePage(), //calls home.dart
    Guidelines(), //calls guidelines.dart
    DiscoverPage(), //calls discover.dart
    AddSamplePage(), //calls add.dart
    BlogsPage(), //calls blogs.dart
    AboutUsPage(), //calls AboutUs.dart
  ];
```

```
  void onTappedBar(int index) {
    setState(() {
//setting state where selected index is index
      _selectedIndex = index;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: _children[_selectedIndex],
      //bottom navigation bar
      bottomNavigationBar: BottomNavigationBar(
        onTap: onTappedBar,
        currentIndex: _selectedIndex,
        items: [
          //BottomNavigationBarItem    is    responsible    to
displaying Icons and label or name of the page
          BottomNavigationBarItem(icon:      Icon(Icons.home),
label: 'Home'),
          BottomNavigationBarItem(
              icon:       Icon(Icons.checklist),       label:
'Guidelines'),
          BottomNavigationBarItem(icon:   Icon(Icons.explore),
label: 'Discover'),
          BottomNavigationBarItem(icon:
Icon(Icons.add_circle), label: 'Add'),
          BottomNavigationBarItem(icon:  Icon(Icons.bookmark),
label: 'Blogs'),
          BottomNavigationBarItem(icon:    Icon(Icons.person),
label: 'AboutUs'),
        ],
        //setting   the   color   scheme   of   the   selected   and
unselected page
        selectedItemColor: Colors.blue,
        unselectedItemColor: Colors.grey,
      ),
    );
  }
}
```

AboutUs.dart: It is contains the code for the About Us navigation Screen.

```
//This code is for the about us page of the app
import 'package:flutter/material.dart';

class AboutUsPage extends StatefulWidget {
  const AboutUsPage({Key? key}) : super(key: key);

  @override
  State<AboutUsPage> createState() => _AboutUsPageState();
}

class _AboutUsPageState extends State<AboutUsPage> {
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      resizeToAvoidBottomInset: false,
      appBar: new AppBar(
        title: Center(child: Text('About Us')),
        backgroundColor: Colors.blue[
            900], //blue[900] is the specific color scheme to
match the color theme of website
      ),
      body: new SingleChildScrollView(
        child: new Center(
          child: new Column(
            children: <Widget>[
              new Image(
                //fetching image from the internet if not
connected the image will not load.
                image: NetworkImage(
                  'https://cdn.searchenginejournal.com/wp-
content/uploads/2022/01/about-us-page-examples-1-
61fd8f9784626-sej-1520x800.jpg',
                ),
                height: 300,
                width: 300,
              ),
              new Container(
                padding: EdgeInsets.all(10),
                child: new Center(
                  child: RichText(
                    text: TextSpan(
                      text: 'About Us',
                      style: TextStyle(
                          height: 1.75,
                          fontSize: 35,
                          color: Colors.blueAccent,
                          fontWeight: FontWeight.bold),
```

```
                    children: <TextSpan>[
                      TextSpan(
                        text: '\n',
                        style: TextStyle(
                            fontWeight:      FontWeight.bold,
color: Colors.white),
                      ),
                      TextSpan(
                        text:
                            "We are a team of faculties and
students working on this research project to discover/innovate
the ways to detect signs of Dysgraphia (Hand-Writing Disorder)
by using an automated system based on Machine Learning, so that
negligence regarding the care needed in this disorder can be
avoided. This will ensure that individuals suffering from this
disorder get timely assistance, and help they need from their
parents, teachers, and society.",
                        style: TextStyle(
                            fontWeight: FontWeight.normal,
                            color: Colors.black,
                            fontSize: 20),
                      ),
                      TextSpan(
                        text: '\n',
                        style: TextStyle(
                            fontWeight:      FontWeight.bold,
color: Colors.white),
                      ),
                      TextSpan(
                        text:
                            "You can also reach out to us with
your queries at our mail: ",
                        style: TextStyle(
                            fontWeight: FontWeight.bold,
                            color: Colors.black,
                            fontSize: 25),
                      ),
                      TextSpan(
                        text: "tild@iiitkota.ac.in",
                        style: TextStyle(
                            fontWeight: FontWeight.bold,
                            color: Colors.blue,
                            fontSize: 25),
                      ),
                    ],
                  ),
                ),
              ),
```

```
                    )
                ],
            ),
        ),
    ),
  );
 }
}
```

Add.dart: This file contains the code for fetching an image either from camera or from gallery and then applying machine learning model on the image.

```
//This code is for the adding a sample and test the image.
//This is the navigation screen where ML model will be called.
import 'dart:convert';
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';
//image picker is used to fetch image from the gallery or camera
import 'package:image_picker/image_picker.dart';
//tflite is used to make our python built ML model work with
flutter app.
//tflite is uses a deprecated version of Android so there maybe
some warning if someone runs this code but the app will be
built.
import 'package:tflite/tflite.dart';
import 'package:http/http.dart' as http;

class AddSamplePage extends StatefulWidget {
  const AddSamplePage({Key? key}) : super(key: key);

  @override
  State<AddSamplePage> createState() => _AddSamplePageState();
}

class _AddSamplePageState extends State<AddSamplePage> {
  //Declaring variables with their types.
  File? image;
  List confidence = [];
  String result = '-1';

//Function for opening gallery and fetching image that needs to
be analysed.
  Future _openGallery(ImageSource source) async {
    final image = await ImagePicker().pickImage(source:
source);
    if (image == null) return;
    final imageTemporary = File(image.path);
//calling do image classification function on the image just
after the task of fetching image.
    setState(() => {this.image = imageTemporary});
    Navigator.of(context).pop();
  }

//Function for opening camera and capture image that needs to
be analysed.
  Future _openCamera(ImageSource source) async {
```

```
// function to call Tflite machine learning model
  doImageClassification(BuildContext context) async {
    try {
      _showProgressBar(context);
      var       postUri       =       Uri.parse("https://api-
dysgraphia.herokuapp.com/predict");

      http.MultipartRequest request =
          new http.MultipartRequest("POST", postUri);

      http.MultipartFile multipartFile =
          await           http.MultipartFile.fromPath('image',
image!.path);

      request.files.add(multipartFile);

      http.StreamedResponse response = await request.send();
      http.Response           res           =           await
http.Response.fromStream(response);
      final Data = jsonDecode(res.body);
      result = Data["dysgraphic"];
      confidence = Data["features"];
      setState(() {
        print(Data["dysgraphic"]);
      });
      Navigator.pop(context);
    } catch (e) {
      Navigator.pop(context);
      print(e);
    }
  }

//widget contains action of Gallery and camera button.
  Future<void> _showChoiceDialog(BuildContext context) {
    return showDialog(
        context: context,
        builder: (BuildContext context) {
          return AlertDialog(
            title: const Text('Make a Choice'),
            content: SingleChildScrollView(
              child: ListBody(
                children: <Widget>[
                  GestureDetector(
                    child: const Text("Gallery"),
                    onTap: () {
                      _openGallery(ImageSource.gallery);
                    },
                  ),
```

```
                const Padding(padding: EdgeInsets.all(8.0)),
                GestureDetector(
                  child: const Text("Open Camera"),
                  onTap: () {
                    _openCamera(ImageSource.camera);
                  },
                ),
              ],
            ),
          ),
        );
      });
}

@override
void initState() {
  // TODO: implement initState
  super.initState();
  //loadModel();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Center(child: Text('Add & Test Sample')),
      backgroundColor: Colors.blue[900],
    ),
    body: Container(
      padding: const EdgeInsets.symmetric(vertical: 16),
      child: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: <Widget>[
            Spacer(),
            image != null
                ? Image.file(
                    image!,
                    width: 350,
                    height: 350,
                    fit: BoxFit.cover,
                  )
                : const Text("No Image Selected"),
            ElevatedButton(
              onPressed: () {
                _showChoiceDialog(context);
              },
              child: const Text('Select Image'),
```

```
                ),
                ElevatedButton(
                    onPressed: () {
                      doImageClassification(context);
                    },
                    child: Text("Analyse Image")),
                Spacer(),
                image != null
                    ? Container(
                        padding:
                            EdgeInsets.symmetric(vertical:    0,
horizontal: 10),
                        child: Row(
                          crossAxisAlignment:
CrossAxisAlignment.center,
                          mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
                          mainAxisSize: MainAxisSize.min,
                          children: [
                            Expanded(
                                child: Text(
                                    'result: ${result == '-1' ?
'' : result == '0' ? 'Non-Dysgraphic' : 'Dysgraphic'}')),
                            SizedBox(width: 16),
                            Expanded(child: Text('Sample Result:
$confidence')),
                          ],
                        ),
                      )
                    : SizedBox.shrink(),
                Spacer(),
                Divider(),
              ],
            ),
          ),
        ),
      );
  }
}
```

Blogs.dart: This file contains the code for the blogs navigation screen.

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
//url launcher is the plugin responsible for routing to a
network link.
import 'package:url_launcher/url_launcher.dart';

class BlogsPage extends StatefulWidget {
  const BlogsPage({Key? key}) : super(key: key);

  @override
  State<BlogsPage> createState() => _BlogsPageState();
}

class _BlogsPageState extends State<BlogsPage> {
  //launch url
  launchURL(String url) async {
    if (await canLaunch(url)) {
      await launch(url, forceWebView: true);
    } else {
      throw 'Could not Launch $url';
    }
  }

  @override
  Widget build(BuildContext context) => Scaffold(
        appBar: AppBar(
          title: Center(child: Text('Read Blogs!')),
          backgroundColor: Colors.blue[900],
        ),
        body: ListView(
          padding: EdgeInsets.all(16),
          children: [
            //calling 3 cards at the momment app only has 3
blogs.
            buildImageInteractionCard(),
            buildImageInteractionCard2(),
            buildImageInteractionCard3()
          ],
        ),
      );
//interaction cards or containers
  Widget buildImageInteractionCard() => Card(
        clipBehavior: Clip.antiAlias,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(24),
        ),
```

```
        child: Column(
          children: [
            Stack(
              children: [
                Ink.image(
                  //network image won't load if not connected
to a network.
                  image: NetworkImage(

'https://images.ctfassets.net/p0qf7j048i0q/3EA8A6B977ED4142819
F5A3CFD502426/90c3053354405bb2fd194d3881ec99e5/i1137021753.jpg
?w=1000&h=750&fit=fill&fm=webp',
                  ),
                  height: 240,
                  fit: BoxFit.cover,
                ),
                Positioned(
                  bottom: 16,
                  right: 16,
                  left: 16,
                  child: Text(
                    'Dysgraphia: How can I help my child?',
                    style: TextStyle(
                      fontWeight: FontWeight.bold,
                      color: Colors.white,
                      fontSize: 24,
                    ),
                  ),
                ),
              ],
            ),
            Padding(
              padding: EdgeInsets.all(16).copyWith(bottom: 0),
              child: Text(
                'A child with dysgraphia has handwriting that
is worse than you would normally see in a child of his age,
intelligence, and education level.',
                style: TextStyle(fontSize: 16),
              ),
            ),
            ButtonBar(
              alignment: MainAxisAlignment.start,
              children: [
                TextButton(
                  child: Text('Learn More'),
                  onPressed: () {
                    const url =
```

```
'https://blog.allaboutlearningpress.com/dysgraphia/';

                    launchURL(url);
                  },
                ),
              ],
            )
          ],
        ),
      );

  Widget buildImageInteractionCard2() => Card(
        clipBehavior: Clip.antiAlias,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(24),
        ),
        child: Column(
          children: [
            Stack(
              children: [
                Ink.image(
                  image: NetworkImage(
                    'https://www.edsys.in/wp-
content/uploads/Dysgraphia-Everything-You-Need-To-Know-1.jpg',
                  ),
                  height: 240,
                  fit: BoxFit.cover,
                ),
                Positioned(
                  bottom: 16,
                  right: 16,
                  left: 16,
                  child: Text(
                    'Dysgraphia: Everything You Need to know',
                    style: TextStyle(
                      fontWeight: FontWeight.bold,
                      color: Colors.black,
                      fontSize: 24,
                    ),
                  ),
                ),
              ],
            ),
            Padding(
              padding: EdgeInsets.all(16).copyWith(bottom: 0),
              child: Text(
```

```
                'One in five students struggles with
dysgraphia, limiting their abilities to use written expression.
Some classroom environments are more successful for students
with dysgraphia. ',
                style: TextStyle(fontSize: 16),
              ),
            ),
            ButtonBar(
              alignment: MainAxisAlignment.start,
              children: [
                TextButton(
                  child: Text('Learn More'),
                  onPressed: () {
                    const url =

'https://www.learningsuccessblog.com/blog-subject/dysgraphia';

                    launchURL(url);
                  },
                ),
              ],
            )
          ],
        ),
      );

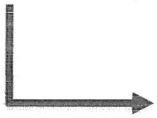
  Widget buildImageInteractionCard3() => Card(
        clipBehavior: Clip.antiAlias,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(24),
        ),
        child: Column(
          children: [
            Stack(
              children: [
                Ink.image(
                  image: NetworkImage(

'https://d1c337161ud3pr.cloudfront.net/files%2F21cdd8d1-a2e2-
4245-9424-f1d892529279_Coping%20with%20Dyspraxia.png',
                  ),
                  height: 240,
                  fit: BoxFit.fitHeight,
                ),
                Positioned(
                  bottom: 16,
                  right: 16,
                  left: 16,
```

```
                    child: Text(
                      'Early and Automated Diagnosis of
Dysgraphia',
                      style: TextStyle(
                        fontWeight: FontWeight.bold,
                        color: Colors.black87,
                        fontSize: 24,
                      ),
                    ),
                  ),
                ],
              ),
              Padding(
                padding: EdgeInsets.all(16).copyWith(bottom: 0),
                child: Text(
                  'Research has demonstrated high similarity
between developmental dyslexia and dysgraphia. Thus, the aim of
the study was to investigate cognitive deficits as potential
predictors of dysgraphia, analogously to those for dyslexia, in
order to identify dysgraphia profiles, depending on the
particular underlying disorder.',
                  style: TextStyle(fontSize: 16),
                ),
              ),
              ButtonBar(
                alignment: MainAxisAlignment.start,
                children: [
                  TextButton(
                    child: Text('Learn More'),
                    onPressed: () {
                      const url = 'https://tild.co.in/';

                      launchURL(url);
                    },
                  ),
                ],
              )
            ],
          ),
        );
}
```

Discover.dart: It is file containing information about steps of how our is working.



Process.dart: It is a part of Discover.dart file, it contains DataModel for the Discover Page. It stores information of each step in the form of an array.

Discover.dart

```
import 'package:flutter/material.dart';
//This file contains information related to each step. In the
form of an array with Description, Image and description.
import 'package:dysgraphia_app1/process.dart';

class DiscoverPage extends StatefulWidget {
  const DiscoverPage({Key? key}) : super(key: key);

  @override
  State<DiscoverPage> createState() => _DiscoverPageState();
}

class _DiscoverPageState extends State<DiscoverPage> {
  //lets us control which page want it to display and setting
its initial state to 0 means 1st page.
  late PageController pageController;
  int _currentPage = 0;

  @override
  void initState() {
    // TODO: implement initState
    super.initState();
    pageController = PageController(initialPage: _currentPage);
  }

  @override
  void dispose() {
    // TODO: implement dispose
    super.dispose();
    pageController.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Center(child: Text('Discover')),
        backgroundColor: Colors.blue[900],
      ),
      resizeToAvoidBottomInset: false,
      body: Padding(
          padding: const EdgeInsets.symmetric(vertical: 40.0),
```

```
            child: Column(
              children: <Widget>[
                const Padding(
                  padding: EdgeInsets.all(10),
                  child: Center(
                    child: Text(
                      "How does this Work?",
                      style: TextStyle(
                          fontWeight: FontWeight.bold,
                          color: Colors.blueAccent,
                          fontSize: 30),
                    ),
                  ),
                ),
                AspectRatio(
                    aspectRatio: 0.85,
                    child: PageView.builder(
                      scrollDirection: Axis.vertical,
                      itemCount: images.length,
                      physics: const ClampingScrollPhysics(),
                      controller: pageController,
                      itemBuilder: (context, index) {
                        return carouselView(index);
                      },
                    ))
              ],
            )),
      );
    }

    Widget carouselView(int index) {
      return carouselCard(images[index]);
    }

  //carousal cards to display images in a styled manner
    Widget carouselCard(DataModel data) {
      return Column(
        children: <Widget>[
          Expanded(
  //this area manages the images that we want to display.
            child: Padding(
              padding: const EdgeInsets.all(30.0),
              child: Container(
                decoration: BoxDecoration(
                    color: Colors.white,
                    borderRadius: BorderRadius.circular(30),
                    image: DecorationImage(
```

```
                        image: AssetImage(data.imageName), fit:
BoxFit.fill),
                    boxShadow: const [
                      BoxShadow(
                          offset: Offset(0, 10),
                          blurRadius: 4,
                          color: Colors.black87)
                    ]),
              ),
            ),
          ),

//this calls title of each page.
          Padding(
            padding: const EdgeInsets.only(top: 50),
            child: Text(
              data.title,
              style: const TextStyle(
                  color: Colors.blueAccent,
                  fontSize: 25,
                  fontWeight: FontWeight.bold),
            ),
          ),

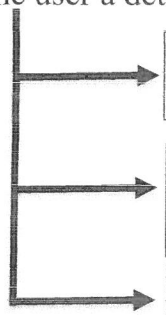
//this area calls the description of the steps from process
          Padding(
            padding: const EdgeInsets.all(8.0),
            child: Text(
              data.desc,
              style: const TextStyle(
                  color: Colors.black, fontSize: 20, fontWeight:
FontWeight.bold),
            ),
          )
        ],
      );
    }
}
```

Process.dart:

```
//this file contains DataModel for the Discover page.
// Which stores information of each step in the form of an array
class DataModel {
  //defining variables and its data type
  final String title;
  final String imageName;
  final String desc;
  //What all information will data model hold.
  DataModel(
    this.title,
    this.imageName,
    this.desc,
  );
}

List<DataModel> images = [
  DataModel("Step-1", 'assets/one.png',
      "Have your child write a sample of handwritten text on A4
size paper as per the guidelines."),
  DataModel("Step-2", 'assets/two.png',
      "The handwriting sample that will be uploaded on our
portal will help us train our model for achieving an impeccable
precision."),
  DataModel("Step-3", 'assets/three.png',
      "Our Machine learning model will then run an DNN based
algorithmic automated analysis of your child's handwriting
sample."),
  DataModel("Step-4", 'assets/four.png',
      "Based on various extracted features and classifying them
across several categories, our model will score the sample to
log the results."),
  DataModel("Step-5", 'assets/five.png',
      "The results are then uploaded on the portal, we intend
to retain the data for further processing to train our model
for every viable cases."),
  DataModel("Step-6", 'assets/six.png',
      "Our team will reach out to you to provide further
assistance, and support."),
];
```

guidelines.dart: This file contains the code of the navigation screen guidelines which gives the user a detail about rules and regulation and guidelines to use the app.



button_widget.dart: This file is associated with guidelines page class, button widget has been used to built button and give functioning to it in this file.

pdf_viewer_page.dart: This file contains the code for controlling the page view while viewing the pdf like going on to the next page using a button or simply scrolling is controlled by this file.

pdfcalling.dart: This file contains the function for calling the pdf or fetching the pdf from the asset folder to be displayed in the Application. It has two functions first loadAsset that is the pdf file for rules, regulations and guidelines, Second storefile which specifies the path where the pdf is stored.

guidelines.dart:

```
//this is one of the main screen where guidelines will be shown
//pdf viewer is the plugin used to display pdf either
// from assets or network or firebase or fetch any pdf to view.
import 'package:dysgraphia_app1/pdf_viewer_page.dart';
//this file contains the functioning of calling a asset pdf file
import 'package:dysgraphia_app1/pdfcalling.dart';
import 'package:flutter/material.dart';
//this file is just for displaying buttons.
import 'button_widget.dart';

class Guidelines extends StatefulWidget {
  const Guidelines({Key? key}) : super(key: key);

  @override
  State<Guidelines> createState() => _GuidelinesState();
}

class _GuidelinesState extends State<Guidelines> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
      appBar: new AppBar(
        title: Center(child: Text('Guidelines')),
        backgroundColor: Colors.blue[900],
      ),
      body: new SingleChildScrollView(
        child: new Center(
          child: new Column(
            children: <Widget>[
              new Container(
```

```
                padding: EdgeInsets.all(10),
                child: new Center(
                  child: RichText(
                    text: TextSpan(
                      text: 'Guidelines',
                      style: TextStyle(
                          height: 1.75,
                          fontSize: 35,
                          color: Colors.blueAccent,
                          fontWeight: FontWeight.bold),
                      children: <TextSpan>[
                        TextSpan(
                          text: '\n',
                          style: TextStyle(
                              fontWeight:    FontWeight.bold,
color: Colors.white),
                        ),
                        TextSpan(
                          text:
                              "A guide to help you with the
process for contributing to our research.",
                          style: TextStyle(
                              fontWeight: FontWeight.bold,
                              color: Colors.black,
                              fontSize: 20),
                        ),
                        TextSpan(
                          text: '\n',
                          style: TextStyle(
                              fontWeight:     FontWeight.bold,
color: Colors.white),
                        ),
                        TextSpan(
                          text: "Major guidelines to keep in
notice are:",
                          style: TextStyle(
                              fontWeight: FontWeight.bold,
                              color: Colors.black,
                              fontSize: 25),
                        ),
                        TextSpan(
                          text: '\n',
                          style: TextStyle(
                              fontWeight:     FontWeight.bold,
color: Colors.white),
                        ),
                        TextSpan(
                          text:
```

```
                                "● Please use an unruled and
blank paper for submitting the sample.\n"
                                "● The recommended dimension of
the paper is A4 (210 x 297 mm). \n"
                                "● Kindly write the text 2 times
(Twice) on the same paper; i.e,.two paragraphs.\n"
                                "● Kindly record the time taken
by your ward to complete the task; You will be asked to submit
the time (in minutes) \n"
                                "● Upload the handwriting sample
photo only in .jpg/.png/.jpeg format\n",
                            style: TextStyle(
                                fontWeight: FontWeight.normal,
                                color: Colors.black,
                                fontSize: 20),
                        ),
                        TextSpan(
                          text: "● The text for the handwriting
sample is:",
                          style: TextStyle(
                              fontWeight: FontWeight.bold,
                              color: Colors.black,
                              fontSize: 20),
                        ),
                      ],
                    ),
                  ),
                ),
              ),
              new Image(
                image: AssetImage('assets/sample.png'),
                // height: 500,
                // width: 500,
              ),
//button widget file is useful here where button is already
built and we need to just use and define its functionality.
              ButtonWidget(
                text: 'Guidelines for Existing Users',
                onClicked: () async {
                  final  path  =  'assets/existing_users.pdf';
//asset file
                  final file = await PDFApi.loadAsset(
                      path); //load asset function defined in
pdf calling file
                  openPDF(context,
```

```
                        file); //calling open pdf function with
parameters file and context
                  },
                  key: null,
                ),
                const SizedBox(height: 16),
                ButtonWidget(
                  text: 'Guidelines for New Users',
                  onClicked: () async {
                    final path = 'assets/new_users.pdf';
                    final file = await PDFApi.loadAsset(path);
                    openPDF(context, file);
                  },
                  key: null,
                ),
              ],
            ),
          ),
        ),
      );
  }

//open pdf function
  void     openPDF(BuildContext     context,     file)     =>
Navigator.of(context).push(
        MaterialPageRoute(
            builder: (context) => PDFViewerPage(
                  file: file,
                  key: null,
                )),
      );
}
```

button_widget.pdf:

```
//this file is associated with guidelines page class
// button widget has been used in guidelines to call pdf
import 'package:flutter/material.dart';

//this class will be called in guidelines page
class ButtonWidget extends StatelessWidget {
  final String text;
  final VoidCallback onClicked;

  const ButtonWidget({
    Key? key,
    required this.text,
    required this.onClicked,
  }) : super(key: key);
//Elevated button to open pdf
  @override
  Widget build(BuildContext context) => ElevatedButton(
        style: ElevatedButton.styleFrom(
          minimumSize: Size.square(50),
        ),
        child: Text(text, style: TextStyle(fontSize: 20)),
        onPressed: onClicked,
      );
}
```

pdfcalling.dart:

```
import 'dart:async';
import 'dart:typed_data';
//file picker to pick or load file from asset this plugin is
also used to fetch files from device
import 'package:file_picker/file_picker.dart';
import 'package:flutter/services.dart';
import 'package:path/path.dart';
import 'package:path_provider/path_provider.dart';
//in case we need to call pdf from the network
import 'package:http/http.dart' as http;

import 'dart:io';

class PDFApi {
  static Future<File> loadAsset(String path) async {
    //load the path
    final data = await rootBundle.load(path);
    //convert the file into byte format
    final bytes = data.buffer.asUint8List();
```

```
    return _storeFile(path, bytes);
  }

  static Future<File> _storeFile(String url, List<int> bytes)
async {
    final filename = basename(url);
    final dir = await getApplicationDocumentsDirectory();
//specifying file path
    final file = File('${dir.path}/$filename');
    await file.writeAsBytes(bytes, flush: true);
    return file;
  }
}
```

pdf_viewer_page.dart:

```
//this page controls the page view if there are multiple pages
it controls the
// functionality like going to a next page or scrolling.
import 'dart:async';
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:flutter_pdfview/flutter_pdfview.dart';
import 'package:path/path.dart';

class PDFViewerPage extends StatefulWidget {
  final File file;

  PDFViewerPage({
    Key? key,
    required this.file,
  }) : super(key: key);

  @override
  _PDFViewerPageState createState() => _PDFViewerPageState();
}

class _PDFViewerPageState extends State<PDFViewerPage> {
  late PDFViewController controller;
  //setting the initial page to 0 means the first page.
  int pages = 0;
  int indexPage = 0;

  @override
  Widget build(BuildContext context) {
    //widget of the page will display name of the pdf
    final name = basename(widget.file.path);
    final text = '${indexPage + 1} of $pages';
```

```
    return Scaffold(
      appBar: AppBar(
        title: Text(name),
        actions: pages >= 2
            ? [
                Center(child: Text(text)),
                IconButton(
                  icon: Icon(Icons.chevron_left, size: 32),
                  onPressed: () {
                    final page = indexPage == 0 ? pages :
indexPage - 1;
                    controller.setPage(page);
                  },
                ),
                IconButton(
                  icon: Icon(Icons.chevron_right, size: 32),
                  onPressed: () {
                    final page = indexPage == pages - 1 ? 0 :
indexPage + 1;
                    controller.setPage(page);
                  },
                ),
              ]
            : null,
      ),
      body: PDFView(
        filePath: widget.file.path,
        // autoSpacing: false,
        // swipeHorizontal: true,
        // pageSnap: false,
        // pageFling: false,
//Switching to the Next page functionality after opening the
pdf
        onRender: (pages) => setState(() => this.pages =
pages!),
        onViewCreated: (controller) =>
            setState(() => this.controller = controller),
        onPageChanged: (indexPage, _) =>
            setState(() => this.indexPage = indexPage!),
      ),
    );
  }
}
```

Home.dart: This file contains the code for the home page of the application.

```
//This is the Home page of the app which displays information
about the cause we are trying to work on.
import 'package:flutter/material.dart';

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      resizeToAvoidBottomInset: false,
      appBar: new AppBar(
        title: Center(child: Text('Home')),
        backgroundColor: Colors.blue[900],
      ),
      body: new SingleChildScrollView(
        child: new Center(
          child: new Column(
            children: <Widget>[
              new Image(
                image: AssetImage('assets/H.png'),
                height: 300,
                width: 300,
              ),
              new Container(
                padding: EdgeInsets.all(5),
                child: new Center(
                  child: RichText(
                    text: TextSpan(
                      text: 'Early and Automated Diagnosis of
Dysgraphia',
                      style: TextStyle(
                          height: 1.75,
                          fontSize: 18,
                          color: Colors.black,
                          fontWeight: FontWeight.bold),
                      children: <TextSpan>[
                        TextSpan(
                          text: '\n',
                          style: TextStyle(
```

```
                                fontWeight:      FontWeight.bold,
color: Colors.white),
                        ),
                        TextSpan(
                          text:
                              "          Leveraging technology
to bring change.",
                          style: TextStyle(
                              fontWeight: FontWeight.bold,
                              color: Colors.blue,
                              fontSize: 16),
                        ),
                      ],
                    ),
                  ),
                ),
              ),
              new Container(
                padding: EdgeInsets.all(5),
                child: new Center(
                  child: RichText(
                    text: TextSpan(
                      text: 'What is Dysgraphia?',
                      style: TextStyle(
                          height: 1.75,
                          fontSize: 35,
                          color: Colors.blueAccent,
                          fontWeight: FontWeight.bold),
                      children: <TextSpan>[
                        // TextSpan(
                        //   text: '\n',
                        //   style: TextStyle(
                        //        fontWeight: FontWeight.bold,
color: Colors.white),
                        // ),
                        TextSpan(
                          text:
                              "Dysgraphia   is   a   learning
condition that manifests itself in difficulties with writing.
It's a neurological condition that affects both children and
adults. Dysgraphia causes people to use wrong wording for what
they're trying to say, and write that are difficult to read.
Approximately 10% of the children are affected by this disorder.
The aetiology of dysgraphia isn't always clear, yet it can occur
after a traumatic experience in adulthood. You can acquire
methods to assist you overcome some of the obstacles that the
disease poses in school and in life once it has been diagnosed.",
                          style: TextStyle(
```

```
                                fontWeight: FontWeight.normal,
                                color: Colors.black,
                                fontSize: 20),
                          ),
                        ],
                      ),
                    ),
                  ),
                ),
                new Container(
                  padding: EdgeInsets.all(5),
                  child: new Center(
                    child: RichText(
                      text: TextSpan(
                        text: '✓ What are we trying to achieve?',
                        style: TextStyle(
                            height: 1.75,
                            fontSize: 35,
                            color: Colors.blueAccent,
                            fontWeight: FontWeight.bold),
                        children: <TextSpan>[
                          TextSpan(
                            text: '\n',
                            style: TextStyle(
                                fontWeight:     FontWeight.bold,
color: Colors.white),
                          ),
                          TextSpan(
                            text:
                                "The objective of this research
project is to discuss the ongoing research and development
towards detecting dysgraphia, by using the given handwriting of
the person that is being tested on by using a set of Machine
Learning/Deep Learning Algorithms and other generic python
libraries.",
                            style: TextStyle(
                                fontWeight: FontWeight.normal,
                                color: Colors.black,
                                fontSize: 20),
                          ),
                        ],
                      ),
                    ),
                  ),
                ),
                new Container(
                  padding: EdgeInsets.all(5),
                  child: new Center(
```

```
                    child: RichText(
                      text: TextSpan(
                        text:  '✓ Bringing changes on all-
fronts.',
                        style: TextStyle(
                            height: 1.75,
                            fontSize: 35,
                            color: Colors.blue,
                            fontWeight: FontWeight.bold),
                        children: <TextSpan>[
                          TextSpan(
                            text: '\n',
                            style: TextStyle(
                                fontWeight:     FontWeight.bold,
color: Colors.white),
                          ),
                          TextSpan(
                            text:
                                "Another objective is to lay out
a foundation on the types of Dysgraphia and their causes. We
are also discussing the importance of detecting this disorder
at an early stage or an early age of the child, and its effects
on   the   overall   growth   and   development   (physically,
psychologically, and physiologically) of an individual.",
                                style: TextStyle(
                                    fontWeight: FontWeight.normal,
                                    color: Colors.black,
                                    fontSize: 20),
                          ),
                        ],
                      ),
                    ),
                  ),
                )
              ],
            ),
          ),
        ),
      );
  }
}
```

We have used plugins in pubspec.yaml file for the development of the application are as follows:

But lets understand what plug-ins in flutter are:

Plug-ins: A piece of software that adds capabilities to your app. For example, you might want your mobile app to interact with the camera on your device. Plugins are an important part of the Flutter ecosystem.

The list of the plugins used in the application build environment are as follows:

- **Cupertino_icons:** This is an asset repo containing default set of icons assets used by flutter Cupertino widgets.
- **url_launcher:** Plugin for launching a URL.
- **drop_shadow_image:** Flutter package for creating drop shadow image with a lot of properties to make it look good.
- **Get:** It helps in route managetment quickly.
- **Tflite:** It helps in integrate machine-learning model with mobile apps.
- **flutter_pdfview:** It provides a PDFView widget.
- **file_picker:** Package that allows you to use a native file explorer to pick single or multiple absolute file paths, with extension filtering support.
- **Path:** String-based **path** manipulation library. All of the **path** operations you know and love, with solid support for Windows, POSIX (Linux and Mac OS X), and the web.
- **path_provider:** A Flutter plugin for finding commonly used locations on the filesystem

Pubspec.yaml:

```
name: dysgraphia_app1
description: A new Flutter project.

# The following line prevents the package from being accidentally published to
# pub.dev using `flutter pub publish`. This is preferred for private packages.
publish_to: 'none' # Remove this line if you wish to publish to pub.dev

# 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: ">=2.16.1 <3.0.0"

# Dependencies specify other packages that your package needs
in order to work.
# To automatically upgrade your package dependencies to the
latest versions
# consider running `flutter pub upgrade --major-versions`.
Alternatively,
# dependencies can be manually updated by changing the version
numbers below to
# the latest version available on pub.dev. To see which
dependencies have newer
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter
  google_fonts: ^2.3.1

  # The following adds the Cupertino Icons font to your
application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  url_launcher: ^6.0.20
  drop_shadow_image:
  get: ^4.6.1
  tflite: ^1.1.2

  #display pdf
  flutter_pdfview:
```

```
  #path providers
  file_picker:
  path:
  path_provider:


dev_dependencies:
  image_picker: 0.8.4+11
  flutter_test:
    sdk: flutter
  # The "flutter_lints" package below contains a set of
recommended lints to
  # encourage good coding practices. The lint set provided by
the package is
  # activated in the `analysis_options.yaml` file located at
the root of your
  # package. See that file for information about deactivating
specific lint
  # rules and activating additional ones.
  flutter_lints: ^1.0.0

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

# 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
assets:
  - assets/
```