Software Design Document

Gaming App Using Pose Detection

Prepared by:

Group 10

Swathy Krishna S	67	swathysouparnika1211@gmail.com
Gayathri Unikuttan	27	gayathriunikuttan@gmail.com
Vignesh P Ajith	69	vigneshpajith@gmail.com
Kiran R	37	kiranraj110601@gmail.com

CONTENTS

1.INTRODUCTION

Physiotherapy is a necessity for empowering the overall health of autistic and cerebral palsy children. We are proposing an immersive technology-based approach for improving the lower limb and fine motor activities with cognitive skills. The Physiotherapy Guide App is designed to be a comprehensive mobile guide for users who require lower limb physiotherapy, providing them with lower limb-oriented games and feedback to help them achieve their health goals. This can help patients to achieve better rehabilitation outcomes, reduce the risk of further injury, and improve overall mobility and functionality of the lower limbs

1.1 Document Purpose

The purpose of this document is to provide a detailed description of the requirements for the development of an app that utilizes pose detection of lower limb. It gives a detailed description of specific features included in the games. It detects lower limb movements like the number of squats, foot movements etc to play different games specified in this document.

1.2 Product Scope

The app is intended for use by individuals who require lower limb physiotherapy, including those recovering from injury or surgery, those with chronic conditions, and those who wish to maintain their overall health and fitness. The app is designed to be user-friendly, responsive, and secure, and is optimized to run smoothly on both iOS and Android devices.

1.3 Intended Audience and Reading Suggestions

Intended audience includes college professors, developers, industry experts, and other stakeholders who may be involved in the development or evaluation of the system. Review the overall design and system description, then focus on specific requirements and any diagrams or visual aids that can help clarify system behaviour.

1.4 References

● IEEE Standards Association. IEEE Std 1016-2009, IEEE Standard for Information

Technology--Systems Design--Software Design Descriptions. IEEE, 2009.

● Martin, Robert C. Clean Architecture: A Craftsman's Guide to Software Structure and

Design. Prentice Hall, 2017.

2.SYSTEM ARCHITECTURE DESIGN

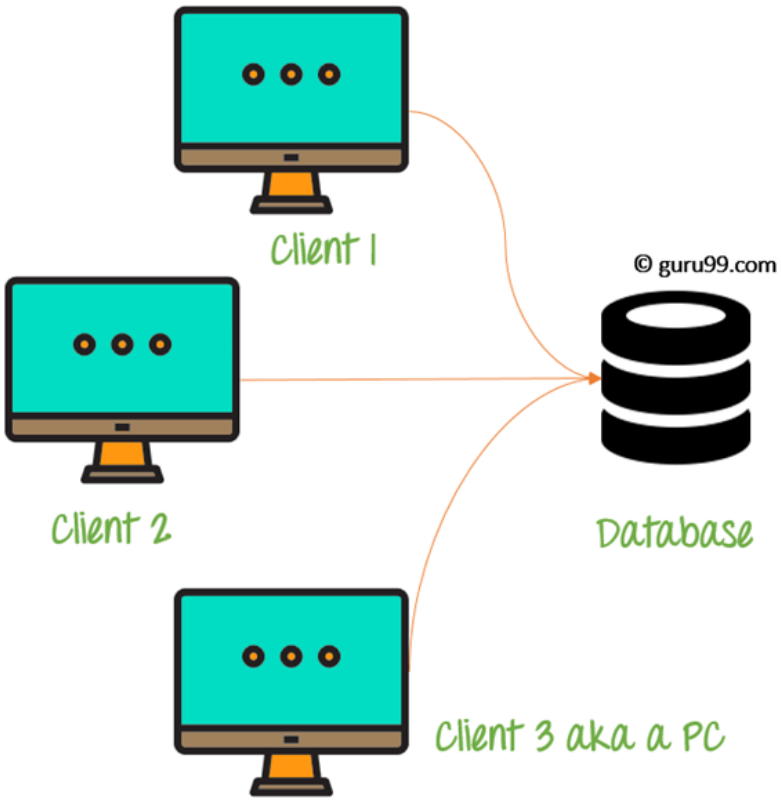
There are various types of architectures available, but in this article we'll focus on the three most popular types, explain how they work, and discuss the benefits they can offer your company.

2.1 Two Tier

Two elements of the game that operate on opposite sides of the design are referred to as a two-tier architecture. The user interface is performed on the client side, which is the first of the two, and database information is stored on the server side.

The business logic and the customer-side application are two web apps that operate on opposite sides of the architecture. Client processes operate on one side of the architecture, while the business application logic can run on either side. This enables the entire application to operate more effectively for the user.

In order to collect data from the user and transmit it back to the database server, the client-side application usually runs on the client computer. This ensures that the two users interact consistently.



2.2 Three Tier

Some companies may benefit from the two-tier architecture's simplicity, but others might require more functionality. Due to this, the three-tier ecommerce architecture contains all the same elements as the two-tier architecture, plus the business side.

Each of the three layers—the presentation layer, the business layer, and the data layer—operates as a distinct module on a different server even though they collaborate to create the overall architecture.

The three-tier method is more effective at gathering data and enhancing decision-making processes than the two-tier architecture. Let's continue and examine each one's operation in more detail.

a) Presentation Layer

The presentation layer is, as the name suggests, the portion that is shown to the client. The user interface and communication layer of the architecture is where customers engage with websites on the front end while the back end programme gathers data and handles requests.

Desktop apps may be written in a variety of languages, depending on the e-commerce platform, but the presentation layer is frequently developed using HTML, CSS, or JavaScript.

b) Game Control Logic

Game control logic is an essential part of any game software architecture design because it manages the game's flow and makes sure that the rules and mechanics are applied properly. For developing game control logic, keep the following points in mind.

Define the game's principles first. This is necessary before designing the control logic for the game. Player movement, scoring, fighting, and other game mechanics fall under this category. You can start creating the game control logic once you have a firm grasp of the guidelines.

Employing a state machine is a typical method for creating game management logic. This entails first specifying the possible game states (such as "menu,"

"gameplay," and "game over") and then specifying the changes between these states based on player actions or other occurrences.

Handle input and output: The logic used to operate the game must be able to handle human input (such as input from a camera or audio device) and give the player the proper feedback. (e.g. updating the game screen, increasing score). To accomplish this, the user interface and graphics engine must be carefully coordinated with the game's control algorithms.

Game events that need to be handled include motion of characters, recharge of lives and stage completion. This is done by the game control logic. This necessitates meticulous coordination with other game elements like the AI system and physics engine.

It is crucial to test and refine the game control logic as with any other element of game development. To make sure the game is entertaining, captivating, and challenging for players, this entails running it through a variety of situations and adjusting as necessary.

The rules of the game, player input and output, game events, and the game's general flow must all be carefully considered when creating the control logic for a game. You can develop game control logic that is interesting, challenging, and enjoyable for players by using a state machine, trying the design, and iterating on it.

c) Data Layer

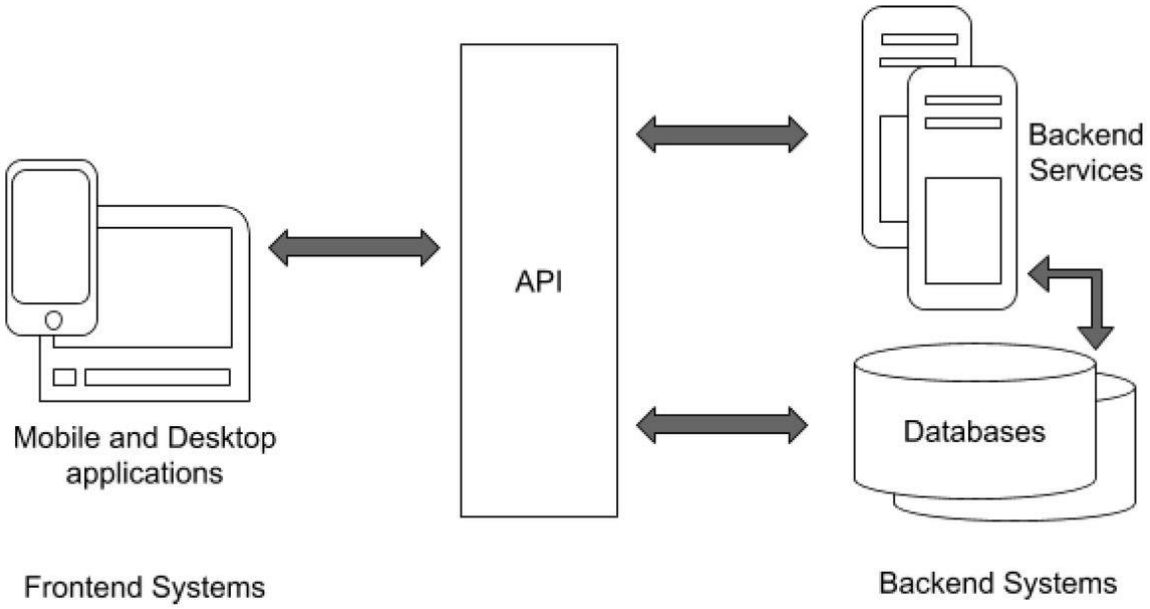
The last layer used to hold data and handle requests is the data tier, also referred to as the database layer. A relational database management system, such as LINQ or SQL, may be used to keep this data.

2.3 Software as a Service

An out-of-the-box e-commerce solution may be more suitable for smaller and/or newer ecommerce companies. Using Software as a Service (SaaS), which is available from different web browsers and stores software and data in the cloud, this type of architecture is possible.

SaaS enables you to launch your website quickly with a complete product catalogue and backend features because the provider manages site upkeep, hosting, and performance.

A SaaS architecture also enables merchants to rapidly update their website to the most recent edition because updates take place in real time. As a result, retailers avoid the hassle of having to change their current options each time the platform is updated.

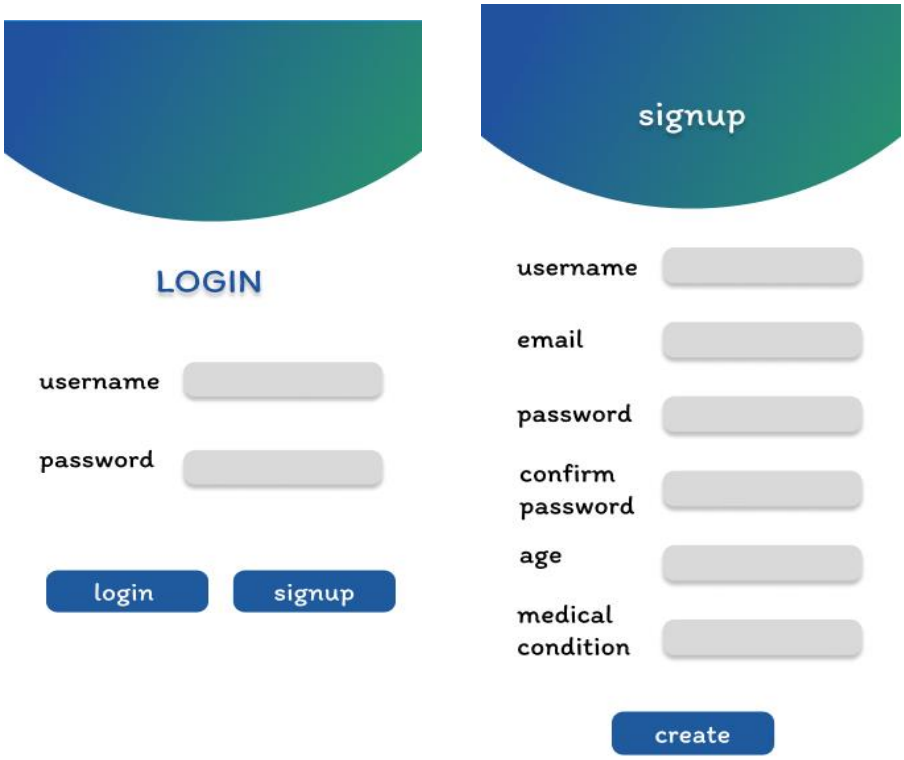


3.Graphical User Interface of The Application

3.1 User Interface Design

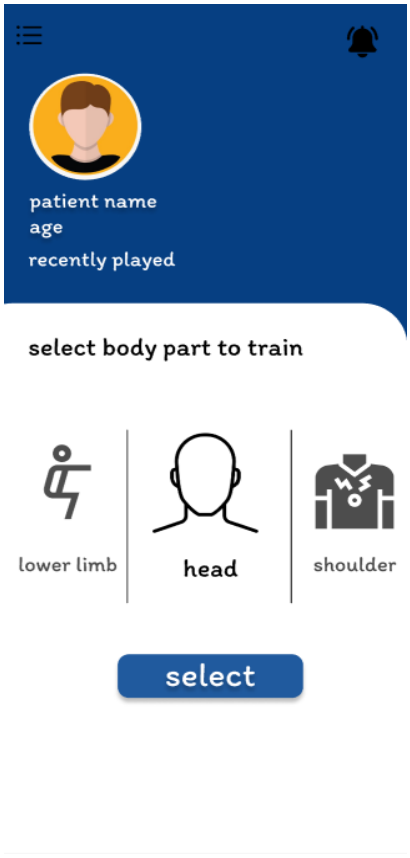
1.Login page and signup page:

The user login page is the first screen that the user sees after opening the app. User can enter username and password and login to their profile. The login page

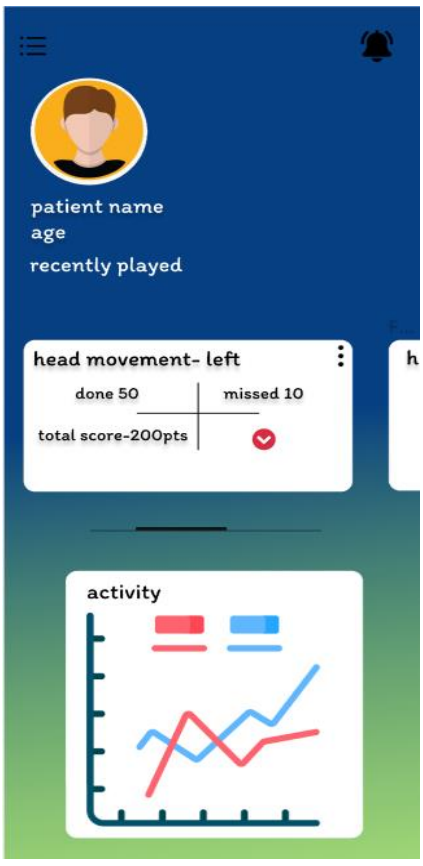


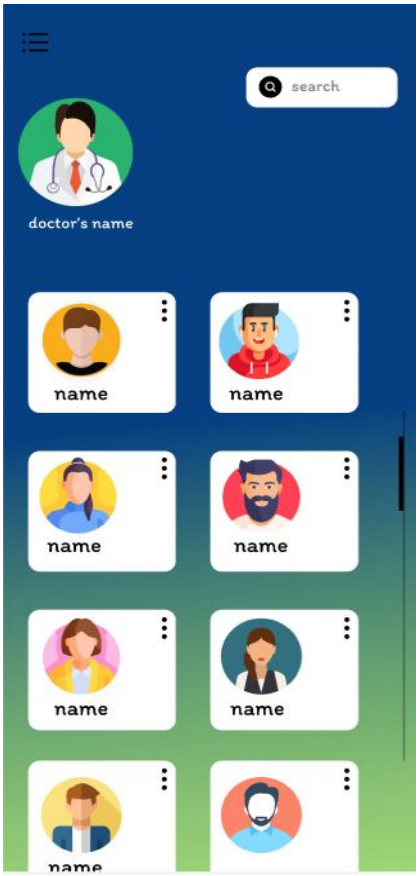
will also have a button which will redirect the user to the signup page of the app. The signup page requires details such as the username, email, password, age and medical condition of the user.

2.Navigation menu: A navigation menu with options to select each of the three games is provided. And a navigation menu to select the difficulty of the games played.

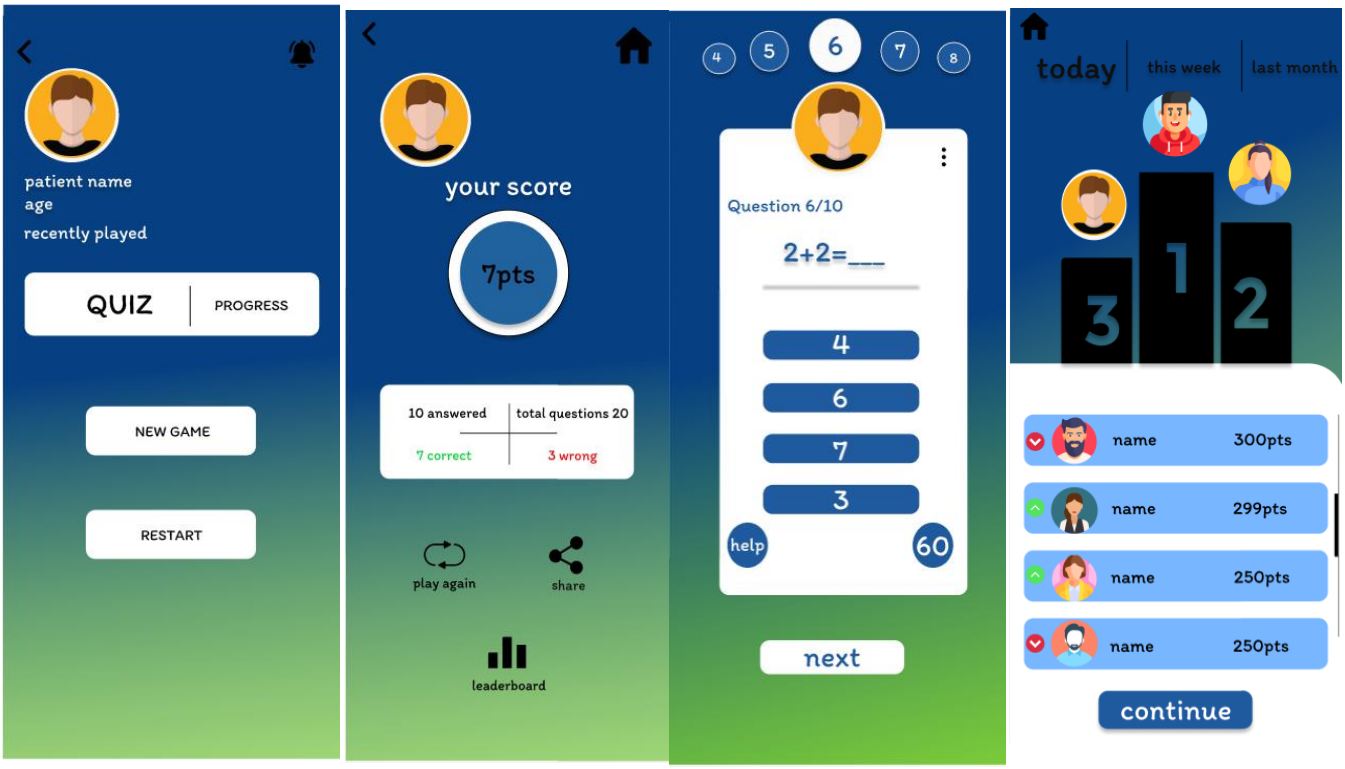


3.User Profile: A user profile picture, username and age are displayed on the top left corner of the screen at each navigation menu.





4.Leaderboard: A leader board page showing the top scores of each game is displayed



5.Game UI:

5.1.Flappy Bird Game Screen:

The Flappy Bird game screen contains the following elements:

Game Title: The title of the game is displayed at the top of the screen.

Game Character: The Flappy Bird character is displayed on the

side of the screen.

Obstacles: Obstacles in the form of pipes are displayed on the right side of the screen.

5.2.Dino Run Game Screen:

The Dino Run game screen contains the following elements:

Game Title: The title of the game is displayed at the top of the screen.

Game Character: The Dino character is displayed on the left side of the screen.

Obstacles: Obstacles in the form of cacti are displayed on the right side of the screen.

5.3.Quiz App Screen:

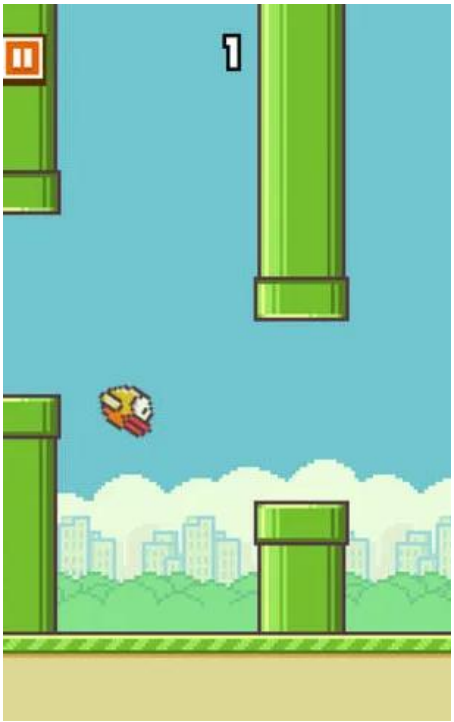
The Quiz App screen contains the following elements:

Game Title: The title of the game is displayed at the top of the screen.

Question: The question is displayed top of the screen

Answer Options: The answer options are displayed on the bottom of the questions.

6.Instructions page: Instructions page will be displayed before every selected game. This page will provide instruction of how to pose for the games.



3.2 Visual Design

- The GUI design will use high-contrast colours and larger font sizes for better visibility and readability.
- The games are made with attractive graphics to grab the attention of autistic children.
- The App is designed to be more visually appealing and engaging for the autistic children.

4.3 Navigation Design

- The application will use a consistent navigation structure throughout.
- The App has easy navigation options that can be simply comprehended by autistic children.

4.TECHNOLOGY STACK

4.1 Google ML Kit (Pose Detection)

ML Kit Pose Detection produces a full-body 33 point skeletal match that includes facial landmarks (ears, eyes, mouth, and nose) and points on the hands and feet. ML Kit's processing happens on-device. This makes it fast and unlocks real-time use cases like processing of camera input. It also works while offline and can be used for processing images and text that need to remain on the device.

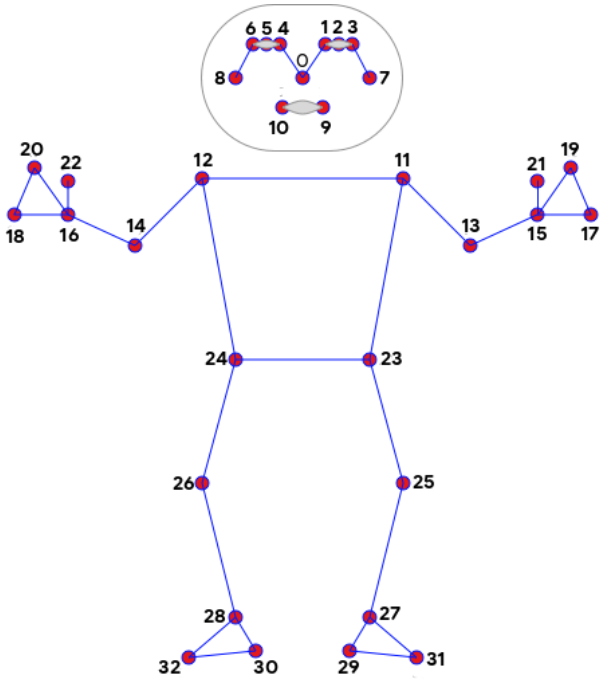
Key capabilities

- **Full body tracking**
- **Cross-platform support**

Z Coordinate for depth analysis

In our app we use the values received from Google ML package to control the game

- Lower limb landmarks (23-32) as shown in following figure are used
- There will be threshold value (different for sensitivity level) and if the movement of the user should be greater than this value to play these games.



Landmarks

0. Nose
1. Left_eye_inner
2. Left_eye
3. Left_eye_outer
4. Right_eye_inner
5. Right_eye
6. Right_eye_outer
7. Left_ear
8. Right_ear
9. Left_mouth
10. Right_mouth
11. Left_shoulder
12. Right_shoulder
13. Left_elbow
14. Right_elbow
15. Left_wrist
16. Right_wrist
17. Left_pinky
18. Right_pinky
19. Left_index
20. Right_index
21. Left_thumb
22. Right_thumb
23. Left_hip
24. Right_hip
25. Left_knee
26. Right_knee
27. Left_ankle
28. Right_ankle
29. Left_heel
30. Right_heel
31. Left_foot_index
32. Right_foot_index

4.2 Google Firebase

It is a mobile application development platform from Google with powerful features for developing, handling, and enhancing applications. Firebase is a backend platform for building web and mobile applications.

Firebase is fundamentally a collection of tools developers can rely on, creating applications and expanding them based on demand.

We have chosen google firebase as our backend, as it is capable of

1. Authentication
2. Cloud Storage
3. Notification

4.3 Flutter

A mobile app development platform that enables the development of high-quality, high-performance mobile applications for both Android and iOS devices. Flutter is Google's SDK for crafting beautiful, fast user experiences for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.

IDE USED: VISUAL STUDIO CODE