**SCIENTIFIC CAREER COUNSELLING SOFTWARE**

A MINI PROJECT REPORT

submitted

*in partial fulfilment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE AND ENGINEERING**

by

**S. SAI KRISHNA 17B81A05L5**

**K. SAI VARDHAN 17B81A05L4**

**D. SAI MALLESH GOUD 17B81A05K0**

Under the guidance of

**D. Shanthi**

**Assistant Professor**

CSE Department



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CVR COLLEGE OF ENGINEERING**

**(*An Autonomous institution, NBA, NAAC Accredited and Affiliated to JNTUH,* *Hyderabad*)**

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),

Rangareddy (D), Telangana- 501 510

June 2020



**Cherabuddi Education Society’s**

**CVR COLLEGE OF ENGINEERING**

**(An Autonomous Institution)**

**ACCREDITED BY NBA, NAAC ‘A’ GRADE (Approved by AICTE & Government of Telangana and Affiliated to JNTU Hyderabad) Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), R.R. District.**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the thesis/ dissertation entitled “**SCIENTIFIC CAREER** **COUNSELLING SOFTWARE**” that is being submitted by **S. SAI KRISHNA (17B81A05L5), K. SAI VARDHAN (17B81A05L4), D. SAI MALLESH GOUD (17B81A05K0)** in partial fulfilment for the award of Bachelor of Technology in ComputerScience and Engineering to the CVR College of Engineering, is a record of bonafide work carried out by them under my guidance and supervision during the year 2020-2021.

The results embodied in this project work has not been submitted to any other University or Institute for the award of any degree or diploma.

|  |  |
| --- | --- |
| Signature of the project guide, | Signature of the HOD |
| **D. Shanthi** | **Dr. K. Venkateswara Rao** |
| Assistant Professor, CSE | Head of Department, CSE |
| CVR College of Engineering | CVR College of Engineering |

**External Examiner**

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is a great pleasure to convey our profound sense of gratitude to our principal

**Dr. Nayanathara K. S**, Vice-Principal **Prof. L. C. Siva Reddy**, **Dr. K. Venkateswara Rao**,Head of CSE Department, CVR College of Engineering, for having been kind enough for arranging necessary facilities for executing the project in the college.

We deem it a pleasure to acknowledge our sense of gratitude to our project guide **Asst. Prof.** **D. Shanthi,** under whom we have carried out the project work. Her incisive and objectiveguidance and timely advice encouraged us with a constant flow of energy to continue the work.

We wish a deep sense of gratitude and heartfelt thanks to the management for providing excellent lab facilities and tools. Finally, we thank all those whose guidance helped us in this regard.

**S. SAI KRISHNA, 17B81A05L5**

**K. SAI VARDHAN, 17B81A05L4**

**D. SAI MALLESH GOUD, 17B81A05K0**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table of Contents** | | | |
|  | | | Page No. |
|  |  | List of Tables | - |
|  |  | List of Figures | - |
|  |  | Symbols | - |
|  |  | Abbreviations | - |
|  |  | List of tables | - |
| 1 |  | **Introduction** | 5 - 6 |
|  | 1.1 | Motivation | 5 |
|  | 1.2 | Problem statement | 6 |
|  | 1.3 | Project report Organization | 6 |
| 2 |  | **Proposed Model** | 7 - 12 |
|  | 2.1 | Introduction to the characteristics of the Problem | 7 |
|  | 2.2 | Design challenges | 10 |
|  | 2.3 | Proposed Solution | 12 |
| 3 |  | **Requirements and Specifications** | 13 - 14 |
|  | 3.1 | Software Requirements | 13 |
|  | 3.1.1 | Functional Requirements | 13 |
|  | 3.1.2 | Non-Functional Requirements | 13 |
|  | 3.2 | System Specifications | 14 |
|  | 3.2.1 | Software Specs | 14 |
|  | 3.2.2 | Hardware Specs | 14 |
| 4 |  | **Analysis and Design** | 15 - 23 |
|  | 4.1 | Use case Diagram | 16 |
|  | 4.2 | Class Diagrams | 17 |
|  | 4.3 | Activity Diagrams | 18 |
|  | 4.4 | Sequence Diagrams | 19 |
|  | 4.5 | System Architecture | 20 |
|  | 4.6 | Technology Description | 21 |
| 5 |  | **Implementation & Testing** | 24 - 44 |
|  | 5.1 | Implementation and Code Snippets | 24 |
|  | 5.2 | Testing | 37 |
|  | 5.3 | Example Screenshots | 42 |
| 6 |  | **Conclusion & Future scope** | 45 |
|  |  | **References** | 46 |
|  |  |  |  |

**CHAPTER – 1 INTRODUCTION:**

**1.1 MOTIVATION**-

Studies that have been done in the past found that there is a relationship between personality types, Knowledge and career choice These studies indicate that there is need to for counsellor to assess the students’ personality type and skill early enough. This will ensure that the students are guided appropriately in their possible career paths and therefore this means that the students will be in a position to select the right subjects. The parents should also be made aware of their children’s uniqueness that exists in their personality in order for them to respect and support these children in their preferences of career choice. Muigai (2007) recommended that a study be done in postsecondary level of education to establish course satisfaction. Despite the presence of school counsellors in most schools, colleges majority of them have teaching responsibilities over and above counselling duties. This dual responsibility leaves the school counsellor with very limited time to provide effective counselling services to the whole population of students. It is on this note that this study is undertaken to establish an effective and efficient way of providing guidance in career matters through the use of technology in order to cater for the needs of a bigger population of students if not all.

**Research Objectives:**

1. To understand the challenges faced by students in choosing careers.
2. To understand the factors that should be considered when guiding students in choosing careers.
   1. To understand the existing tools and methods used by students in career guidance.
   2. To design, develop and test a mobile application that provides career guidance based on test scores.
3. To validate that the application provides accurate and reliable career guidance.

**User Expectations-**

Wants an Ideal Career Guidance-

The desire to have someone suggesting a suitable career.

Wants Development-

The desire to have work opportunities to learn new skills or increase abilities.

Is Enthusiastic-

The tendency to be eager and excited toward one’s own goals.

Wants Self-Improvement-

The tendency to attempt to develop or better oneself.

Wants Advancement-

The desire to have work opportunities to expand one’s career or responsibilities.

Wants Challenge-

The willingness to attempt difficult tasks or goals.

**1.2 PROBLEM STATEMENT-**

The objective is to develop a subjective, psychometric and aptitude test software for different levels of students like school, graduates and dropout students applying artificial intelligence tools, which will help to identify the current status of students and help them for future career counselling purpose. Questionnaires Are be prepared using existing research papers, many acknowledged sites and a well-designed survey to understand types of questions to be asked to students for understanding their basic Personality and skills they possess. A software tool is to be developed to evaluate the aptitude of the students respectively and provide a proper career guidance for their future.

**Keywords**: Career, Career guidance, Personality.

**1.3 ORGANIZATION OF REPORT-**

This chapter puts across the problem that this study is solving, a brief description of what is already happening locally and globally with regards to the field of study in form of an introduction, the objectives of the study motivation, justification of the study and user’s expectation. In the following chapter we will be looking into our Proposed Model of career guidance.

**CHAPTER 2- PROPOSED MODEL:**

**2.1 THE CHARACTERISTICS OF THE PROBLEM-**

It is a comprehensive, developmental program designed to assist individuals in making and implementing informed educational and occupational choices. A career guidance and counselling program develops an individual's competencies in self-knowledge, educational and occupational exploration, and career planning.

Career guidance and counselling programs help individuals acquire the knowledge, skills, and experience necessary to identify options, explore alternatives and succeed in society.

Why Should You Take Career Assessment Tests?

Taking assessment tests will help you in identifying: -

**1. Your Passions and Interests-**

The Internet is full of free self-assessment test that can help you discover your passions and interests some of which you may not be aware of prior to taking the test. Once you have identified your passions and interests it will be easier to make more informed decisions regarding what career path you should choose.

**2. Your Strengths and Weaknesses-**

Some assessment tests are designed to help the quiz takers gain insight into how they operate. This information can help in determining the good traits as well as the bad traits you possess which in turn will help not just in your professional, but in your personal life as well.

**3. Your Marketable Skills-**

Employers are interested in knowing what you can offer to them as an employee that will benefit the business or create some sort of a value for them. Being a fresher with zero work experience, you may not be aware of what skills you possess that might attract the recruiters during campus placements. Taking skill assessment tests will help you gain insights into this aspect of yours. Once you know what you are good at, it becomes easier to convince the recruiters of your worth.

**4. Your Values-**

There are certain assessment tests that are specifically designed for helping the quiz taker realize the values that hold the most importance to them. In other words, these tests help in defining what motivates you and what’s important to you. As values are a lot more personal to every individual, such tests may not as clear-cut as the ones designed for identifying passions and interests, but they do come in very handy when it comes to making career choices.

**5. Your Ideal Profession-**

As fresh job seekers, you may also find yourself in a dilemma as to what career would be the perfect choice for you. Taking various assessment tests will reveal a number of professions that can be good options based on your personality traits, skills, values, passions, and interests.

**6. Set objective for Greater Results-**

A substantial piece of the career advisor’s job is to direct you through the procedure of goal setting in your career endeavours. The professional advisor can enable you to recognize the means expected to achieve vital objectives en-route as you investigate new alternatives or roll out improvements in the profession you as of now have.

**7. Discards confusion-**

With the help of career guidance, the confusion of taking up the right subject decreases and students are able to choose their piece of interests, pick the appropriate subjects, and make the right choice in their career. Career guidance helps the students to make better decisions and have a clear idea about their desired studies and move forward in life. It is really important to be clear in terms of career because once you made your decision, it will stay along with you throughout your life.

**8. Get to know the availabilities-**

There are times when we are unaware of many career options that are available these days; Which is why, career guidance broadens their career options and helps students come across many desired career opportunities like available colleges, admission eligibilities, college insights etc.

**9. Support and motivation-**

As you start working with career counselling, you may encounter many careers that require specific training or education. This process of counselling will enable you to gain the special skillset which is required to fill in for the career of your choice. In addition to this, career advisors always ensure to support and lift the morale of the individual by getting to know what they desire to achieve in future and offer guidance in accordance to that.

**10. Job search support-**

For individuals who are on the lookout for new job or opting out of a career to choose another one, career counselling can help them in such research. They encourage and support the individuals by providing them with interviewing services, cover letter, resume, along with advisory services. They offer feedback tools as well as the resources that are required for the right job hunt and help you land the best dream job ever.

**Drawbacks of taking Career Guidance-**

1. Student must genuinely give the aptitude test otherwise the app will not provide appropriate results.
2. Uncertainty- You may choose to get a degree in accounting but find out after graduation that the job doesn't satisfy you. The job market may change drastically before you graduate. You may not be able to find a position in your chosen field.
3. Multiple solutions which eventually leads to confusion.

* Moreover, we wanted to create a very nice interactive UI where the student will not get irritated and we will be introducing some games too, so that to entertain themselves while taking this test.
* We maintain student data securely as we allow access to Authenticated users only.

**The existing models:**

* Most of them are like the mazes i.e., they will show you the path only if you take their pre-arranged paths, where as ours is like a forest, which ever path you choose, we will guide you through it.
* And also, the data they have is manually labelled without proper experiments.
* Moreover, all those apps don’t consider personality assessment of the student.
* The majority of the Existing models are based on the static data and they have a very small range of careers offered to the use.

**2.2 DESIGN CHALLENGES-**

The identified challenges are:

1.Data collection

2.Data Pre-processing

3.Model selection

4.Data training

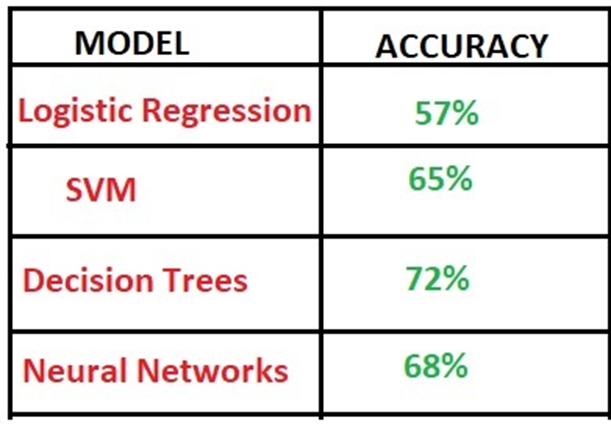
5.Testing

6.Selecting mobile and web frame works

7.Selecting database

8.Selecting cloud platform for deployment

* Data collection is the key part of the entire project, although there were very few datasets related to career and career guidance, we explored a lot and found a Dataset from Kaggle which was suitable and we also added some of the data from our survey.
* But the raw data collected is very noisy. We pre-processed the data by doing feature engineering, ml techniques and finally performed one-hot encoding of data.
* Model selection is also an important part for accuracy of any project. But due to lack of data, upon several trails we choose DECISION TREES as it gave better accuracy for 20% of testing data.



* We saved the best model in. pkl format. And then we manually tested on many test cases. It has shown better results.
* For mobile application we thought to make it available for both iOS and android, if we follow the traditional methodology, we need to make two different applications like android app using android studio and swift for iOS. but we thought using modern platform application kit like flutter Framework where we can deploy it in android, iOS, and even web with the same code. Even though it is new platform we took it as a challenge.
* As we are dealing with ml project. We thought of using python language because it is most suitable for machine learning projects. So, we chose flask-python as backend server which opened many opportunities for flexible development of project.
* We cannot directly upload data to database as we are maintaining it at server, so we made rest api request and response to interact with user and server. As we need to store text data for each user, for saving space and faster response, we chose SQLite database.
* We want a remote server rather than local host, so we thought of deploying it in a cloud platform. After quite a research we found Heroku as a faster, cheap and reliable cloud platform which has preinstalled OS and has preinstalled python add-ons, also deployment is very easy in it as we just need to link our git repository to Heroku master.
* Moreover, we have completed this app in this corona pandemic through zoom calls, git repository for updating our work. It was quite a challenge.

**2.3 PROPOSED SOLUTION-**

Instead of relying on the static data and the previous market trends like the existing models available in the market, we tried to apply dynamic Artificial Intelligence techniques on the collected user data by the questionnaires on different subjects, personality traits etc. And unlike the existing specific models, this would give a more precise Career guidance as we even consider the logical, personal, subjective knowledge of the individual. From the reviewed literature, it is clear that the students experience a challenge of inadequate career guidance and the need for a system that will address this issue. With such an application, questions testing on personality, subjective knowledge of an individual and also aptitude tests are going to help students narrow down to careers fit for them.

1. Technologies used are Flutter ui sdk, which support dart language (version- 2.8) and Flask micro web framework which supports python (version- 3.7)
2. We have used Waterfall development lifecycle.
3. Decision trees are exactly suitable for this type of data where we need to determine at each step of the process, like if he is good at maths, then we check whether he is good at statistics which so. on brings it to data scientist, or not good in maths then we check for other thing.
4. We used default entropy criteria “Gini index” for controlling the split of decision trees at each node.
5. We evaluated our project based on accuracy of the prediction of the career for certain individual scores as it the key part of our app.
6. Heroku is a container-based cloud Platform as a Service (PaaS). Heroku is used to deploy, manage, and scale modern apps. this platform is elegant, flexible, and easy to use.

**CHAPTER 3- SOFTWARE REQUIREMENT SPECIFICATIONS:**

**3.1 SOFTWARE REQUIREMENTS-**

This section outlines Functional and Non-Functional requirements of the system.

**3.1.1 Functional Requirements-**

These requirements define the capabilities and functions that the implemented system must have in order to achieve its intended purpose. They include:

i. Login and Logout – to gain access to the web application users must login using username and password. Users need to logout in order to exit the system.

* 1. Add professional profile – the system should allow a user who is logged into the system to create a personal profile.
  2. View profile – A user can view their own personal profile stored in the database.

1. Modify records – The administrator should be able to modify career, user and professionals’ records.
2. Take personality test – The student should be able to take a personality test on the mobile application and view their score.
3. View career possibilities – The student should see career possibilities once they get their score

**3.1.2 Non-functional Requirements-**

These requirements that specify the criteria to judge the operation of system. They were constructed in agreement with functional requirements that define specific behaviour and functions. They include:

i. Usability – the system interface should be easy to use.

1. Reliability and availability – the system should be reliable and always available to perform tasks requested by the user.
2. Scalability – the system should be able to adopt additional functionalities. Additional data should be easy to incorporate.
   1. Integrity – the system being data oriented, it should ensure that the data analysed and stored is not altered or corrupted.

v. Performance – the system should have an acceptable response time while performing its functions.

1. Security – The system should allow only authorized users to use its functionalities.

**3.2 SYSTEM SPECIFICATIONS-**

**3.2.1 Software requirements-**

The software used for the development of this project are:

Front end:

Flutter

Back end:

Flask

Sqlachemy

Sklearn

Database:

SQLite

Cloud Deployment:

Heroku

Tools:

Visual studio code

Sublime text

DB browser for SQLite

**3.2.2 Hardware requirements-**

The hardware used for the development of this project are:

1. An Pc of windows OS with a minimum of 8GB RAM.
2. An Android phone with Android Marshmallow OS (6.0.1) or higher along with an USB cable.

**CHAPTER 4- ANALYSIS AND DESIGN:**

Unified Modelling Language (UML) notion was used for modelling and designing diagrams to offer a clear picture of the system to be developed (Object Management Group, 2015). The study employed four different UML diagrams for its design. These diagrams include a use case diagram, database schema, sequence diagram and class diagram.

1. Use Case Diagram- Use cases were used to identify and separate system functionalities in terms of who is responsible for it, thus coming up with actors and uses cases. The actors of this system are students, professionals and the administrator. The use case is in text for describing the action performed by the actors on the system.

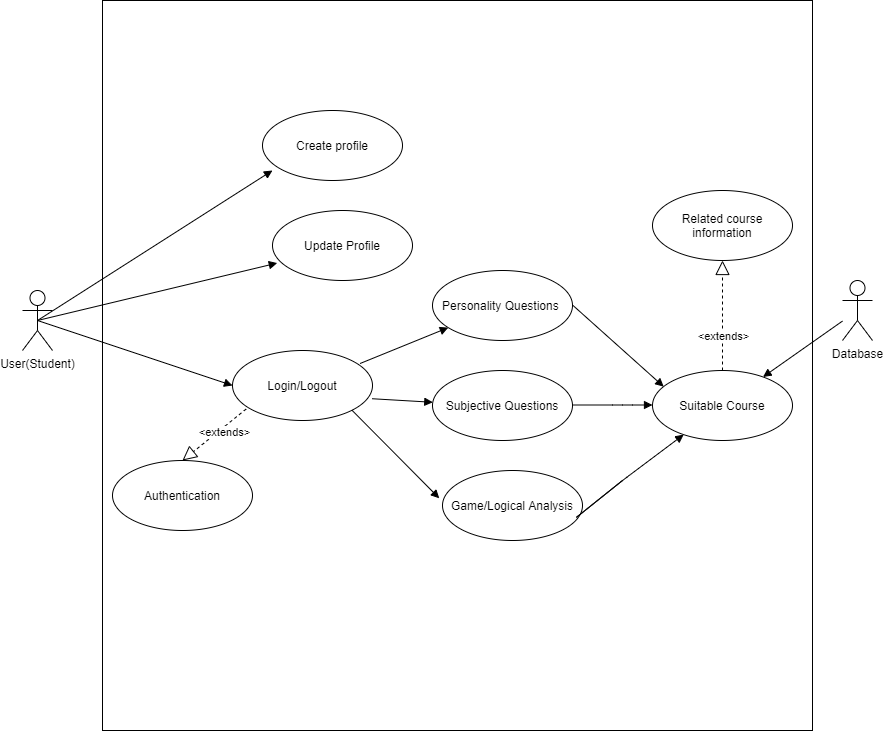
2. Sequence Diagram- The sequence diagram was used to show interaction between the objects. This gives a clear picture of how the system flows from one point to another.

3. Class Diagram- This was used to show objects in the system, their attributes and methods. The relationships between the objects and their cardinality values.

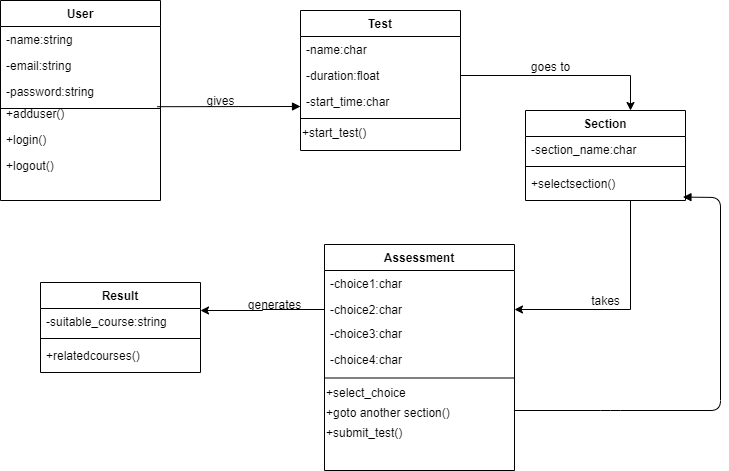
4. Activity diagram- It is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The basic purpose of activitydiagrams is to capture the dynamic behaviour of the system.

5. System Architecture Diagram- A system architecture diagram would be used to show the relationship between different components. Usually they are created for systems which include hardware and software and these are represented in the diagram to show the interaction between them.

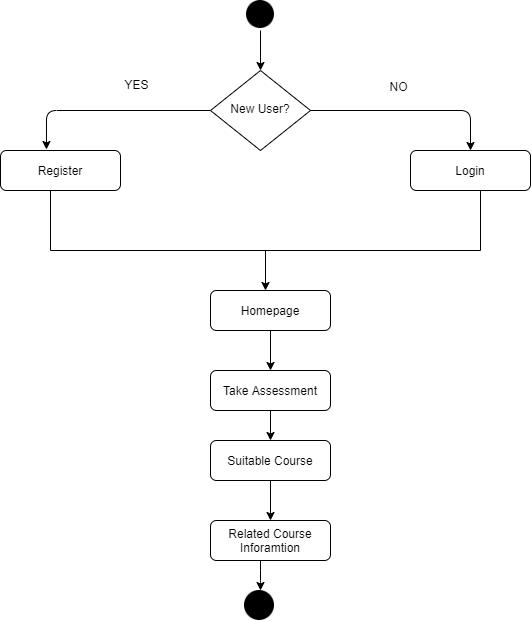
**4.1 USE-CASE DIAGRAM-**



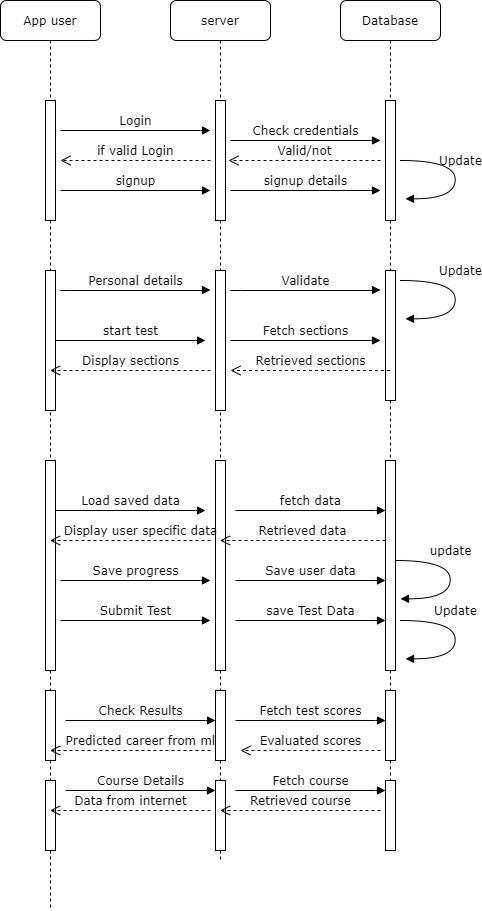
**4.2 CLASS DIAGRAM:**



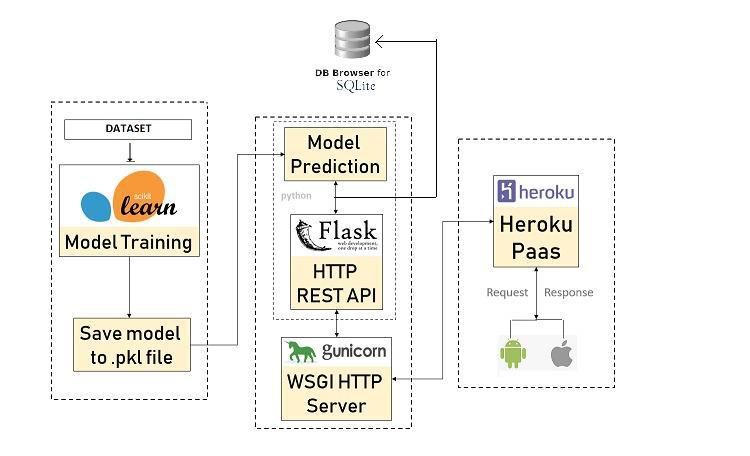
**4.3 ACTIVITY DIAGRAM-**



**4.4 SEQUENCE DIAGRAM-**



**4.5 SYSTEM ARCHITECTURE DIAGRAM-**



**4.6 TECHNOLOGY DESCRIPTION-**

**Flutter SDK –**

Flutter is Google’s UI toolkit for building beautiful, natively compiled applications for [mobile,](https://flutter.dev/docs) [web](https://flutter.dev/web), and [desktop](https://flutter.dev/desktop) from a single codebase. Flutter apps are written in the [Dart](https://en.wikipedia.org/wiki/Dart_(programming_language)) language and make use of many of the language's more advanced features. Flutter runs in the Dart virtual machine which features a [just-in-time](https://en.wikipedia.org/wiki/Just-in-time_compilation) execution engine.

Flutter's engine, written primarily in [C++,](https://en.wikipedia.org/wiki/C%2B%2B) provides low-level [rendering](https://en.wikipedia.org/wiki/Rendering_(computer_graphics)) support using Google's [Skia](https://en.wikipedia.org/wiki/Skia_Graphics_Engine) graphics library. Additionally, it interfaces with [platform-specific](https://en.wikipedia.org/wiki/Platform-specific_model) [SDKs](https://en.wikipedia.org/wiki/Software_development_kit) such as those provided by [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) and [iOS](https://en.wikipedia.org/wiki/IOS). The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers will interact with Flutter via the Flutter Framework, which provides a modern, reactive framework, and a rich set of platform, layout and foundation widgets.

A [Hello, World](https://en.wikipedia.org/wiki/%22Hello,_World!%22_program) program in Flutter looks like this:

1. **import** 'package: flutter/material.dart';
2. void main() => runApp (HelloWorldApp ());
3. **class HelloWorldApp extends** StatelessWidget {
4. @override
5. Widget build(BuildContext context) {
6. **return** MaterialApp(
7. title: 'Hello World App',
8. home: Scaffold(
9. appBar: AppBar(
10. title: Text('Hello World App'),
11. ),
12. body: Center(
13. child: Text('Hello World'),
14. ),
15. ),
16. );
17. }
18. }

**Heroku**-

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

Heroku is fully managed, giving developers the freedom to focus on their core product without the distraction of maintaining servers, hardware, or infrastructure. The Heroku experience provides services, tools, workflows, and polyglot support—all designed to enhance developer productivity.

**Android Studio-**

Android Studio is the official [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) [Google](https://en.wikipedia.org/wiki/Google)'s [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [operating system](https://en.wikipedia.org/wiki/Operating_system), built on [JetBrains'](https://en.wikipedia.org/wiki/JetBrains) [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA) software and designed specifically for [Android development.](https://en.wikipedia.org/wiki/Android_software_development) It is available for download on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) based operating systems. It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (ADT) as the primary IDE for native Android application development.

The following features are provided in the current stable version:

* 1. [Gradle](https://en.wikipedia.org/wiki/Gradle)-based build support
  2. Android-specific [refactoring](https://en.wikipedia.org/wiki/Code_refactoring) and quick fixes
  3. [Lint](https://en.wikipedia.org/wiki/Lint_(software)) tools to catch performance, usability, version compatibility and other problems
  4. [ProGuard](https://en.wikipedia.org/wiki/ProGuard_(software)) integration and app-signing capabilities
  5. Template-based wizards to create common Android designs and components
  6. A rich [layout editor](https://en.wikipedia.org/wiki/Graphical_user_interface_builder) that allows users to drag-and-drop UI components, option to [preview layouts](https://en.wikipedia.org/wiki/WYSIWYG) on multiple screen configurations
  7. Support for building [Android Wear](https://en.wikipedia.org/wiki/Android_Wear) apps
  8. Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
  9. Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of [IntelliJ](https://en.wikipedia.org/wiki/IntelliJ) (and [CLion](https://en.wikipedia.org/wiki/CLion)) e.g. [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [C++,](https://en.wikipedia.org/wiki/C%2B%2B) and more with extensions, such as [Go](https://en.wikipedia.org/wiki/Go_(programming_language)); and Android Studio 3.0 or later supports [Kotlin.](https://en.wikipedia.org/wiki/Kotlin_(programming_language)) External projects [backport](https://en.wikipedia.org/wiki/Backporting) some Java 9 features. While IntelliJ that Android Studio is built on supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

**SQLite**-

SQLite is an in-process library that implements

a [selfcontained,](https://www.sqlite.org/selfcontained.html) [serverless,](https://www.sqlite.org/serverless.html) [zero-configuration,](https://www.sqlite.org/zeroconf.html) [transactional](https://www.sqlite.org/transactional.html) SQL database engine. The code for SQLite is in the [public domain](https://www.sqlite.org/copyright.html) and is thus free for use. SQLite is the [most widely deployed](https://www.sqlite.org/mostdeployed.html) database in the world with more applications than we can count, including several [high-profile projects.](https://www.sqlite.org/famous.html) SQLite is an embedded SQL database engine. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file.

**Flask-**

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

**Gunicorn-**

The Gunicorn "Green Unicorn" is a Python Web Server Gateway Interface HTTP server. It is a pre-fork worker model, ported from Ruby's Unicorn project. The Gunicorn server is broadly compatible with a number of web frameworks, simply implemented, light on server resources and fairly fast.

**5 IMPLEMENTATION AND TESTING-**

Code Snippets-

**5.1 IMPLEMENTATION-**

**5.1.1 Question and answer page-**

import 'package:flutter/rendering.dart';

import 'package:flutter\_custom\_clippers/flutter\_custom\_clippers.dart';

import 'package:flutter/material.dart';

import 'package:miniproject1/style/theme.dart' as Theme;

import 'questions.dart';

import 'home.dart';

// import 'package:path/path.dart';

class sectionA extends StatefulWidget {

  int itemname;

  sectionA(this.itemname);

   int first = 1;

  @override

  \_sectionAState createState() => \_sectionAState();

}

String img;

int \_groupValue = -1;

String str;

int cat = -1;

int item = 0;

bool isdoublecollapsed = false;

bool iscollapsed = false;

final len = 13;

final items = [0, 1, 2, 3, 4, 6, 7, 8, 9,10,11,12];

class \_sectionAState extends State<sectionA> {

  PageController \_controller =

      PageController(initialPage: 0, viewportFraction: 1.0);

  @override

  void dispose() {

    \_controller.dispose();

    super.dispose();

  }

  @override

  Widget build(BuildContext context) {

    cat = widget.itemname;

    print(cat);

    if (widget.first == 1) {

      str = question[cat][0];

      widget.first = 0;

      print("str");

    print(str);

    }

    double c\_width = MediaQuery.of(context).size.width \* 0.8;

    //List<AnimatedPositioned> c = [mycard1(context, 0,true),mycard1(context, 1,true),mycard1(context,1,true)];

    return Scaffold(

      body: Stack(

        children: [

          Stack(

            children: <Widget>[

              ClipPath(

                clipper: WaveClipperTwo(),

                child: Container(

                  decoration: BoxDecoration(

                    gradient: new LinearGradient(

                        colors: [

                          Theme.Colors.loginGradientStart,

                          Theme.Colors.loginGradientEnd

                        ],

                        begin: const FractionalOffset(0.0, 0.0),

                        end: const FractionalOffset(1.0, 1.0),

                        stops: [0.0, 1.0],

                        tileMode: TileMode.clamp),

                  ),

                  child: SingleChildScrollView(

                    scrollDirection: Axis.vertical,

                    child: Column(

                      children: <Widget>[

                isdoublecollapsed

                   ?  Container(

                      padding: EdgeInsets.only(top: 10.0, left: 20, right: 10),

                      alignment: Alignment.topCenter,

                      child:Column(

                        children: <Widget>[

                          Image.network(img,width: 300,height: 150,),

                          SizedBox(height: 10,),

                          Text(str ,style: TextStyle(

                                  color: Colors.black,

                                  fontSize: 15.0,

                                  fontWeight: FontWeight.bold),),

                        ],

                      ))

                      : Container(

                      padding: EdgeInsets.only(top: 100.0, left: 20, right: 10),

                      alignment: Alignment.topCenter,

                      child: iscollapsed

                          ? Image.network(str)

                          : Text(

                              str,

                              style: TextStyle(

                                  color: Colors.black,

                                  fontSize: 20.0,

                                  fontWeight: FontWeight.bold),

                            ),

                      //child:  Text(str, overflow: TextOverflow.ellipsis, style: TextStyle(color: Colors.black ,fontSize: 28.0,fontWeight: FontWeight.bold),),

                    ),

                      ],

                    ),

                  ),

                  height: 300,

                ),

              ),

              // Stack(

              //    children: <Widget>[

              PageView(

                controller: \_controller,

                //scrollDirection: Axis.horizontal,

                physics: const NeverScrollableScrollPhysics(),

                // pageSnapping: false,

                children: <Widget>[

                  mycard1(context, 0),

                  mycard1(context, 1),

                  mycard1(context, 2),

                  mycard1(context, 3),

                  mycard1(context, 4),

                  mycard1(context, 5),

                  mycard1(context, 6),

                  mycard1(context, 7),

                  mycard1(context, 8),

                  mycard1(context, 9),

                ],

              )

              //  ],

              //  children: items.map((item) {

              //             //iscollapsed = true;

              //            return c[item];

              //         }).toList()

              // )

            ],

          ),

        ],

      ),

    );

  }

  Widget mycard1(context, item1) {

    print('hello');

    return new Container(

      padding: EdgeInsets.only(top: 300, left: 30, right: 30, bottom: 100),

      child: new Card(

        color: Colors.transparent,

        elevation: 8.0,

        child: new Container(

          width: (MediaQuery.of(context).size.width) / 1.2,

          height: (MediaQuery.of(context).size.height) / 1.8,

          decoration: new BoxDecoration(

            color: Colors.white,

            borderRadius: BorderRadius.circular(12.0),

          ),

          child: SingleChildScrollView(

            scrollDirection: Axis.vertical,

            child: new Column(

              children: <Widget>[

                new Container(

                  width: (MediaQuery.of(context).size.width) / 1.2,

                  height: (MediaQuery.of(context).size.height) / 1.5 -

                      (MediaQuery.of(context).size.height) / 2.6,

                  padding: EdgeInsets.only(left: 50.0, top: 30.0),

                  child: optionCard(context, item1),

                ),

                new Container(

                  width: (MediaQuery.of(context).size.width) / 1.2,

                  height: (MediaQuery.of(context).size.height) / 1.5 -

                      (MediaQuery.of(context).size.height) / 2.2,

                  child: new Row(

                    mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                    children: <Widget>[

                      new ButtonTheme(

                        height: 50,

                        minWidth: 120,

                        child: RaisedButton(

                          elevation: 16.0,

                          color: Colors.red,

                          shape: RoundedRectangleBorder(

                              borderRadius: BorderRadius.circular(22.0)),

                          hoverColor: Colors.redAccent,

                          child: new Text("prev",

                              style: new TextStyle(

               new ButtonTheme(

                        height: 50,

                        minWidth: 120,

                        child: new RaisedButton(

                          elevation: 16.0,

                          shape: RoundedRectangleBorder(

                              borderRadius: BorderRadius.circular(22.0)),

                          color: Colors.green,

                          hoverColor: Colors.greenAccent,

                          child: new Text("next",

                              style: new TextStyle(

                                  fontSize: 16.0, color: Colors.white)),

                          onPressed: () {

                            setState(() {

}

);

}

}

  Widget optionCard(context, item1) {

    return SingleChildScrollView(

      scrollDirection: Axis.horizontal,

      child: new Column(

        crossAxisAlignment: CrossAxisAlignment.start,

        mainAxisAlignment: MainAxisAlignment.spaceEvenly,

        children: <Widget>[

          // new SizedBox(

          //   height: 20.0,

          // ),

          new Row(

            children: <Widget>[

              new Radio(

                  value: 0,

                  focusNode: null,

                  groupValue:  selected[cat][item1],

                  onChanged: (T) {

                    setState(() {

                      selected[cat][item1] = T;

                      print(T);

                      \_groupValue = T;

                    });

                  }),

              Text(

                option[cat][item1][0],

                style: TextStyle(fontWeight: FontWeight.bold, fontSize: 20.0),

              ),

            ],

          ),

          new Row(

            children: <Widget>[

              new Radio(

                  value: 1,

                  focusNode: null,

                  groupValue:  selected[cat][item1],

                  onChanged: (T) {

                    setState(() {

                      selected[cat][item1] = T;

                      print(T);

                      \_groupValue = T;

                    });

                  }),

              Text(option[cat][item1][1],

                  style:

                      TextStyle(fontWeight: FontWeight.bold, fontSize: 20.0)),

            ],

          ),

          new Row(

            children: <Widget>[

              new Radio(

                  value: 2,

                  groupValue: selected[cat][item1],

                  onChanged: (T) {

                    setState(() {

                      selected[cat][item1] = T;

                      print(T);

                      \_groupValue = T;

                    });

                  }),

              Text(option[cat][item1][2],

                  style:

                      TextStyle(fontWeight: FontWeight.bold, fontSize: 20.0)),

            ],

          ),

          new Row(

            children: <Widget>[

              new Radio(

                  value: 3,

                  focusNode: null,

                  groupValue:  selected[cat][item1],

                  onChanged: (T) {

                    setState(() {

                      selected[cat][item1] = T;

                      print(T);

                      \_groupValue = T;

                    });

                  }),

              Text(option[cat][item1][3],

                  style:

                      TextStyle(fontWeight: FontWeight.bold, fontSize: 20.0)),

            ],

          ),

        ],

      ),

    );

  }

}

**5.1.2 Homepage-**

import 'package:flutter/material.dart';

import 'package:miniproject1/personaldetails.dart';

import 'package:miniproject1/questions.dart';

import 'package:miniproject1/result.dart';

import 'package:miniproject1/starttest1.dart';

import 'package:miniproject1/ui/login\_page.dart';

import 'crop.dart';

import 'package:miniproject1/style/theme.dart' as Theme;

import 'package:path/path.dart';

import 'package:http/http.dart' as http;

import 'instruction.dart';

import 'dart:async';

import 'dart:convert';

import 'package:animated\_text\_kit/animated\_text\_kit.dart';

// class homepage1 extends StatelessWidget

// {

//   String dummy;

//   // This widget is the root of your application.

//   @override

//   Widget build(BuildContext context) {

//     return MaterialApp

//     (

//       home: mainHome(),

//     );

//   }

// }

List selected;

class Tag {

 int zero,one,two,three,four,five,six,seven,eight,nine;

  Tag({this.zero,this.one,this.two,this.three,this.four,this.five,this.six,this.seven,this.eight,this.nine});

  factory Tag.fromJson(dynamic json) {

    return Tag(zero:json['0'],one:json['1'],two:json['two'],three:json['3'],four:json['4'],five:json['5'],six:json['6'] ,seven:json['7'] ,eight:json['8'],nine:json['9']);

  }

  @override

  String toString() {

    return '{ ${this.zero},${this.one},${this.two},${this.three},${this.four},${this.five},${this.six},${this.seven},${this.eight},${this.nine}, }';

  }

}

String snam;

class mainHome extends StatefulWidget {

  String sname;

 mainHome(this.sname);

  @override

  \_mainHomeState createState() => \_mainHomeState();

}

String uri = "https://previews.123rf.com/images/dizanna/dizanna1604/dizanna160400113/54662375-career-word-cloud-with-magnifying-glass-business-concept.jpg";

class \_mainHomeState extends State<mainHome> {

  String currentProfilePic = "https://www.upwork.com/profile-portraits/c1no3FNq0c-o5frTV0fDnJuQTrHt5PD8\_I38\_mnEcaJUBXIsAUPa\_5hCQgZHh36znt";

  String otherProfilePic = "https://yt3.ggpht.com/-2\_2skU9e2Cw/AAAAAAAAAAI/AAAAAAAAAAA/6NpH9G8NWf4/s900-c-k-no-mo-rj-c0xffffff/photo.jpg";

  bool \_isEditingText = false;

  String s = 'sai krishna';

  String profile;

  TextEditingController s1;

  @override

  void initState(){

    super.initState();

    s1 = TextEditingController(text:s);

  }

  @override

  void dispose(){

    s1.dispose();

    super.dispose();

  }

  Widget \_edittext(){

    if(\_isEditingText)

    return Center(

      child:TextField(

        onSubmitted: (newValue){

          setState(() {

            s = newValue;

            \_isEditingText = false;

          });

        },

          autofocus: true,

          controller: s1,

      ),

    );

    return InkWell(

      onTap: (){

        setState(() {

          \_isEditingText = true;

        });

      },

    );

  }

  @override

  Widget build(BuildContext context) {

    snam = widget.sname;

    return new Scaffold(

      appBar: new AppBar(title: new Text("Home Page"), backgroundColor: Colors.blueAccent,),

      drawer: new Drawer(

        child: new ListView(

          children: <Widget>[

            Container(

              width: double.infinity,

              padding: EdgeInsets.all(50.0),

              decoration: BoxDecoration(image: new DecorationImage(image: NetworkImage(

                "https://img.freepik.com/free-photo/3d-grunge-room-interior-with-spotlight-smoky-atmosphere-background\_1048-11333.jpg?size=626&ext=jpg"

                //"https://media.istockphoto.com/photos/blue-abstract-background-or-texture-picture-id1138395421?k=6&m=1138395421&s=612x612&w=0&h=bJ1SRWujCgg3QWzkGPgaRiArNYohPl7-Wc4p\_Fa\_cyA="

               //"https://venngage-wordpress.s3.amazonaws.com/uploads/2018/09/Colorful-Geometric-Simple-Background-Image.jpg"

              ),

              fit: BoxFit.fill,

              )),

              child: Center(

                child: Column(

                  children: <Widget>[

                    GestureDetector(

                    child:CircleAvatar(

                      backgroundImage: NetworkImage("https://www.upwork.com/profile-portraits/c1no3FNq0c-o5frTV0fDnJuQTrHt5PD8\_I38\_mnEcaJUBXIsAUPa\_5hCQgZHh36znt"),radius: 50.0 ,

                    ),

                    onTap: (){

                      Text('hello world');

                    },

                    ),

                    SizedBox(height: 10.0,),

                     TyperAnimatedTextKit(

                       isRepeatingAnimation: false,

                      onTap: (){

                        setState(() {

                           s = 'sunny';

                        });

                      },

                       text: [

                         snam,

                       ],

                       speed: Duration(milliseconds: 150),

                       textStyle: TextStyle(color: Colors.red,fontWeight: FontWeight.bold, fontSize: 20.0,fontFamily: "Bobbers"),

                       textAlign: TextAlign.center,

                       alignment: AlignmentDirectional.topCenter,

                     )

                    // Container(

                    //   width: 100,

                    //   height: 100,

                    //   decoration: BoxDecoration(

                    //     shape: BoxShape.circle,

                    //     border: new Border.all(

                    //       color: Colors.brown,

                    //       width:2.0

                    //     ),

                    //     image: DecorationImage(

                    //       image: NetworkImage(currentProfilePic),

                    //     fit: BoxFit.fill),

                    //   ),

                    // )

                  ],

                ),

            ),

            ),

            // new UserAccountsDrawerHeader(

            //   accountEmail: new Text("sai.sanniboina@gmail.com"),

            //   accountName: new Text("Saikrishna"),

            //   currentAccountPicture: new GestureDetector(

            //     child: new CircleAvatar(

            //       backgroundImage: new NetworkImage(currentProfilePic),

            //       radius :50.0

            //     ),

            //     onTap: () => print("This is your current account."),

            //   ),

            //   decoration: new BoxDecoration(

            //     image: new DecorationImage(

            //       image: new NetworkImage("https://img00.deviantart.net/35f0/i/2015/018/2/6/low\_poly\_landscape\_\_the\_river\_cut\_by\_bv\_designs-d8eib00.jpg"),

            //       fit: BoxFit.fill

            //     )

            //   ),

            // ),

            SizedBox(height: 20,),

            new ListTile(

              title: new Text("Home",style: TextStyle(fontWeight: FontWeight.bold,fontSize: 17),),

              leading: new Icon(Icons.home),

              onTap: () {

                Navigator.of(context).pop();

                //Navigator.of(context).push(new MaterialPageRoute(builder: (BuildContext context) => new Page("First Page")));

                Navigator.push(context, MaterialPageRoute(builder: (context) =>  mainHome(snam)));

              }

            ),

             new ListTile(

              title: new Text("Results",style: TextStyle(fontWeight: FontWeight.bold,fontSize: 17),),

              leading: new Icon(Icons.announcement),

              onTap: () {

                Navigator.of(context).pop();

                //Navigator.of(context).push(new MaterialPageRoute(builder: (BuildContext context) => new Page("First Page")));

                getresults(context);

              }

            ),

            new ListTile(

              title: new Text("settings",style: TextStyle(fontWeight: FontWeight.bold,fontSize: 17),),

              leading: new Icon(Icons.settings),

              onTap: () {

                Navigator.of(context).pop();

                Navigator.push(context, MaterialPageRoute(builder: (context) =>  MyApp()));

               // Navigator.of(context).push(new MaterialPageRoute(builder: (BuildContext context) => new Page("MyApp")));

              }

            ),

          ],

        ),

      ),

      body: Stack(

        children:<Widget>[

           Padding(

             padding: const EdgeInsets.only(top:58.0,left: 5,right: 5),

             child: Container(

              decoration: BoxDecoration(

                borderRadius: BorderRadius.circular(8.0),

                image:DecorationImage(image: NetworkImage(uri,),

                fit: BoxFit.cover

              ),

          ),

          ),

           ),

          //child:Text('hello')

      Padding(

        padding: EdgeInsets.only(top:25,left:90),

        child:

        GestureDetector(

      child:Container(

          decoration: BoxDecoration(

            borderRadius: BorderRadius.circular(10.0),

                    gradient: new LinearGradient(

                        colors: [

                          Theme.Colors.loginGradientStart,

                          Theme.Colors.loginGradientEnd

                        ],

                        begin: const FractionalOffset(0.0, 0.0),

                        end: const FractionalOffset(1.0, 1.0),

                        stops: [0.0, 1.0],

                        tileMode: TileMode.clamp),

          ),

                    //shape: RoundedRectangleBorder(borderRadius: BorderRadius.all(Radius.circular(5.0))),

                    child: Padding(

                      padding: const EdgeInsets.symmetric(vertical: 10.0, horizontal: 42.0),

                      child: Text(

                        "start test",

                        style: TextStyle(

                            color: Colors.black,

                            fontSize: 25.0,                            fontFamily: "WorkSansBold"),

                      ),

                    ),

      ),

                    onTap: ()

              {

                getselecteddata();

                getpersonpage(context);

// Navigator.of(context).push(MaterialPageRoute(builder: (context) => UserLogin(snam)));

                        }

          ),

      ),

        ],

      ),

    );

  }

**5.2 TESTING-**

**5.2.1 Testing on connection between flask and flutter app rest api connection-**

Future getresults(context) async{

     var response = await http.post(

    address+'/getresult',

    headers: <String, String>{

      'Content-Type': 'application/json; charset=UTF-8',

    },

    body: jsonEncode(<String, String>{

      'user' : snam,

    }),

  );

  print(response.statusCode);

  var jsondata = json.decode(response.body);

  String p = jsondata['output'];

  Navigator.push(context, MaterialPageRoute(builder: (context) => Resultpage(p)));

  }

    Future getpersonpage(context) async

  {

    var response = await http.post(

    address+'/personpage',

    headers: <String, String>{

      'Content-Type': 'application/json; charset=UTF-8',

    },

    body: jsonEncode(<String, String>{

      'user' : snam,

    }),

  );

  print(response.statusCode);

  print(response.body);

  var jsondata = json.decode(response.body);

  String p = await jsondata['output'];

  if(p == 'done'){

    Navigator.of(context).push(MaterialPageRoute(builder: (context) => instruct(snam)));

  }

  else{

    Navigator.of(context).push(MaterialPageRoute(builder: (context) => UserLogin(snam)));

  }

  }

  Future getselecteddata() async

  {

    var response = await http.post(

    address+'/getsavedprogress',

    headers: <String, String>{

      'Content-Type': 'application/json; charset=UTF-8',

    },

    body: jsonEncode(<String, String>{

      'user' : snam,

    }),

  );

  print(response.statusCode);

  print(response.body);

  print(response.body.runtimeType);

  var jsondata = json.decode(response.body);

  String p = jsondata['output'];

  List L = p.split('|');

  List<List> sub = [];

  List<List> sel1 = [];

  for(int i = 0;i<L.length;i++){

    List<String> k = L[i].split(',');

    sub.add(k);

  }

  print(sub);

  for(int i = 0;i<sub.length;i++)

  {

      List<int> dataListAsInt = sub[i].map((data) => int.parse(data)).toList();

      sel1.add(dataListAsInt);

  }

  print(sel1);

  // List c = jsonDecode(p);

  // print(c);

  // var tagObjsJson = json.decode(response.body)['output'] as List;

  // print(tagObjsJson);

  // print("########");

  // var tagObjs = tagObjsJson.map((tagJson) => Tag.fromJson(tagJson)).toList();

  // print(tagObjs);

  // print(tagObjs.runtimeType);

  setState(() {

    selected = sel1;

  });

  }

}

**5.2.2 Testing on authentication-**

Future logindata(context) async{

  http.Response response = await http.post(

    address+'/login',

    headers: <String, String>{

      'Content-Type': 'application/json; charset=UTF-8',

    },

    body: jsonEncode(<String, String>{

      'username': loginEmailController.text,

      'password': loginPasswordController.text

    }),

  );

  print(response.statusCode);

  var jsondata = json.decode(response.body);

  String x = jsondata['Lstatus'];

  x = x.toString();

  print(x);

  if(x == "success")

  {

  Navigator.of(context).push(MaterialPageRoute(builder: (context) => mainHome(loginEmailController.text),));

 }

else{

showInSnackBar(context,x);

    }

}

Future registerdata(context) async{

    http.Response response = await http.post(

    address+'/register',

    headers: <String, String>{

      'Content-Type': 'application/json; charset=UTF-8',

    },

    body: jsonEncode(<String, String>{

      'username': signupNameController.text,

      'password': signupPasswordController.text

    }),

  );

  print(response.statusCode);

  var jsondata = json.decode(response.body);

  String x = jsondata['Rstatus'];

  x = x.toString();

  print(x);

  if(x == "done")

  {

  Navigator.of(context).push(MaterialPageRoute(builder: (context) => mainHome(signupNameController.text),));

 }

**5.2.3 Testing on evaluation of score-**

Widget score (List allselected){

    correct = [];

     for(int i = 0;i<13;i++)

     {

       int count = 0;

       List x = allselected[i];

       for(int j = 0;j<10;j++)

       {

         if(answer[i][0] == 'E'){

           String ans = answer[i][j+1];

           if(ans.length > 1){

              var finalans = ans.split(',');

              String a = finalans[0];

              String b = finalans[1];

              if (x[j] == op[a] || x[j] == op[b])

              {

                count += 1;

              }

           }

           else{

              if (x[j] == op[ans])

              {

                count += 1;

              }

             }

         }

         else{

         if(x[j] == op[answer[i][j]])

         {

            count += 1;

         }

         }

       }

       correct.add(count);

     }

     correct.add(\_research);

     correct.add(90);

     String cor = correct.toString();

     print('###########');

     print(cor);

    senddata(cor);

  }

   senddata(String title) async{

    print(title);

   var response = await http.post(

    address+'/predict',

    headers: <String, String>{

      'Content-Type': 'application/json; charset=UTF-8',

    },

    body: jsonEncode(<String, String>{

      'title': title,

      'user':user

    }),

  );

  print(response.statusCode);

  var jsondata = json.decode(response.body);

  print(jsondata['output']);

  setState(() {

    fop = jsondata['output'];

  });

  fop = fop.toString();

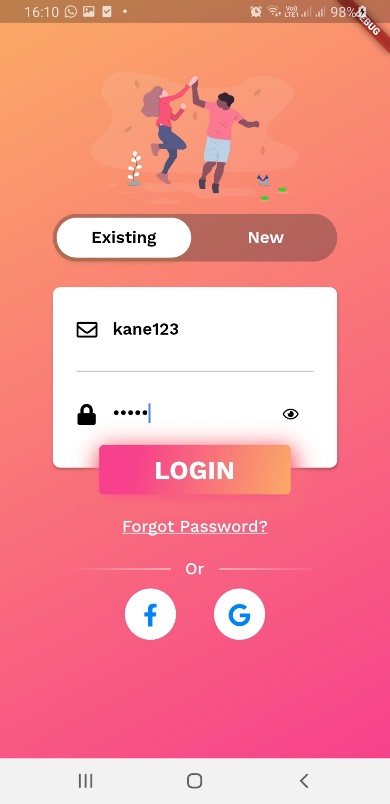
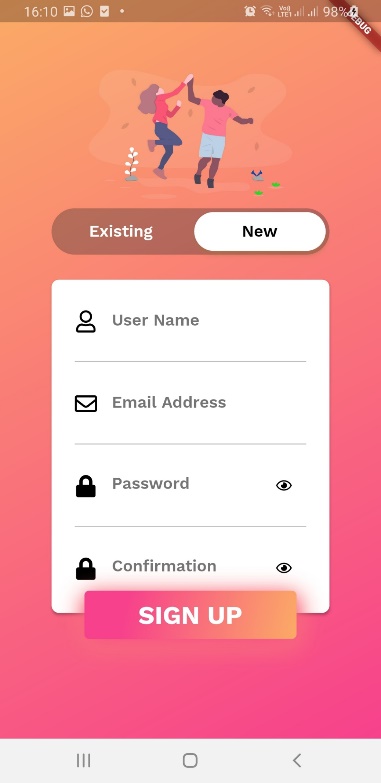
   Navigator.of(context).push(MaterialPageRoute(builder: (context) => Resultpage(fop)));

}

**5.3 EXAMPLE SCREENSHOTS-**

* Signup page and Login Page-

A Signup page (also known as a registration page) enables users and organizations to independently register and gain access to our app.

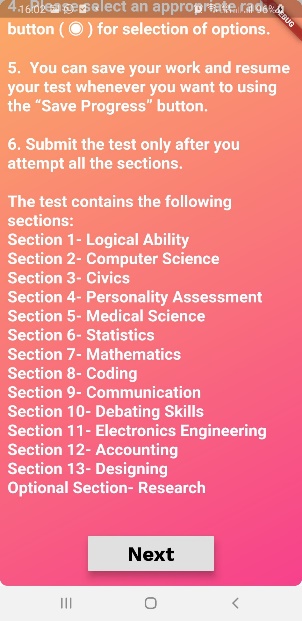
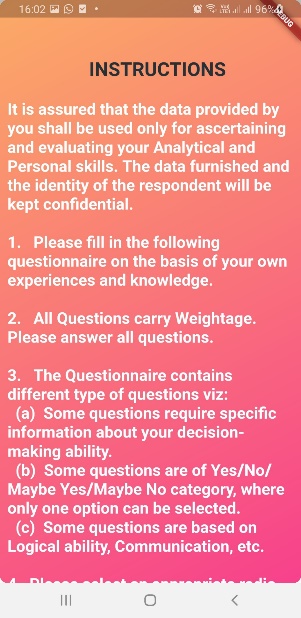
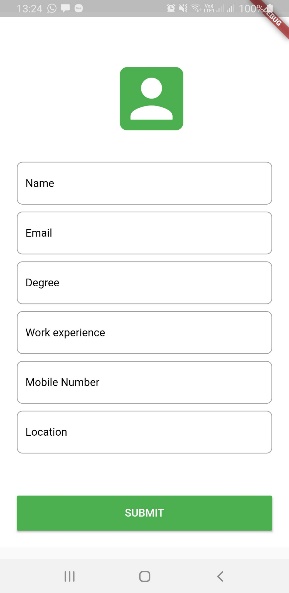


* Personal details Page-

Information of the User such as his/her Name, Email, Education, etc.

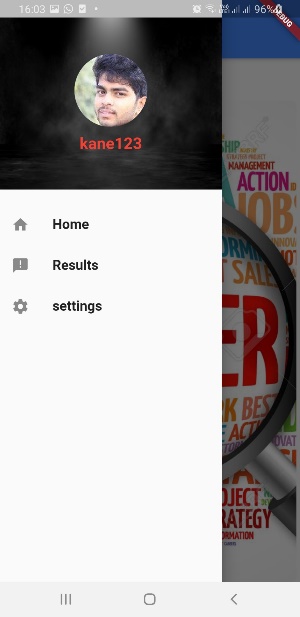
And Instruction Page contains the steps or the order that should be followed in order to

attempt the Assessment of our App.



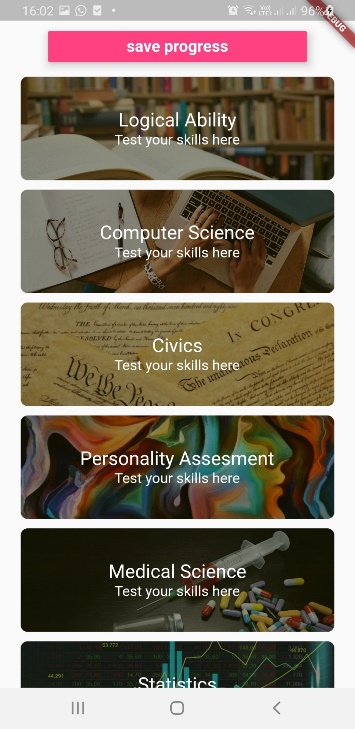
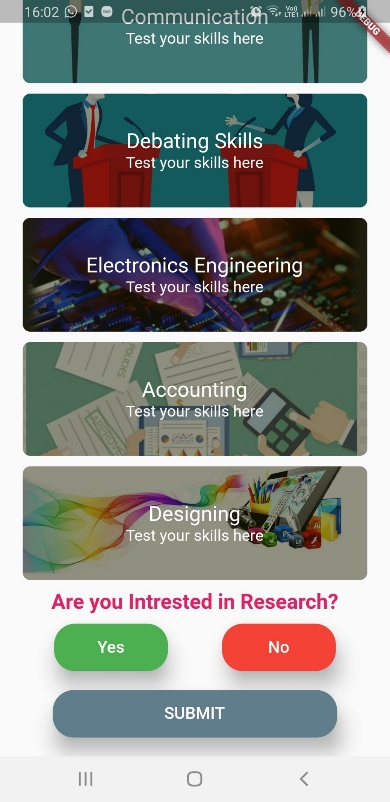
* Homepage**-**

A homepage is a webpage that serves as the starting point of the app. It is the default webpage that loads when you login into the app.



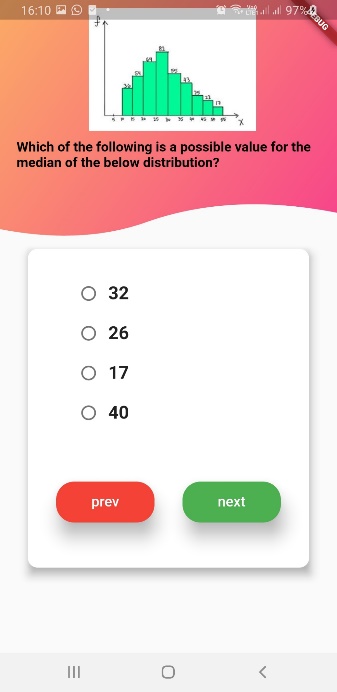
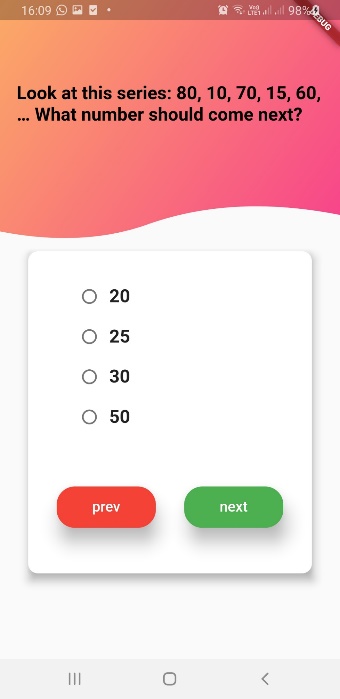
* Sections Page-

This pagecontains all the sections which a User can choose into for the Assessment in our App.

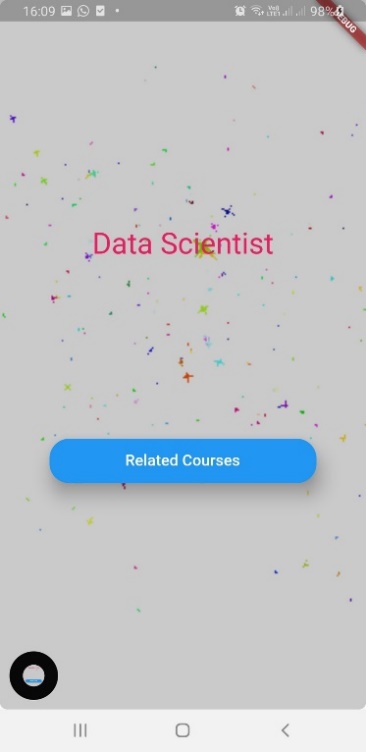
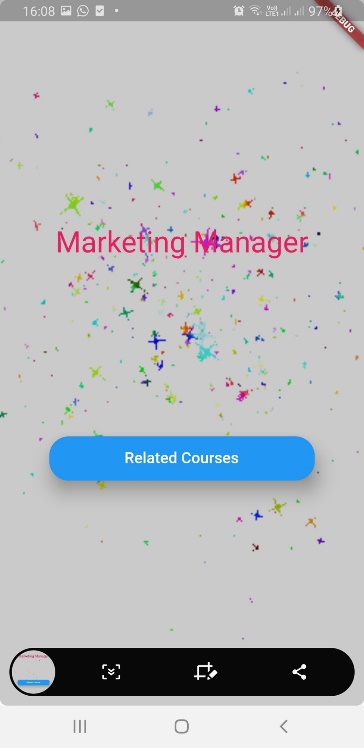
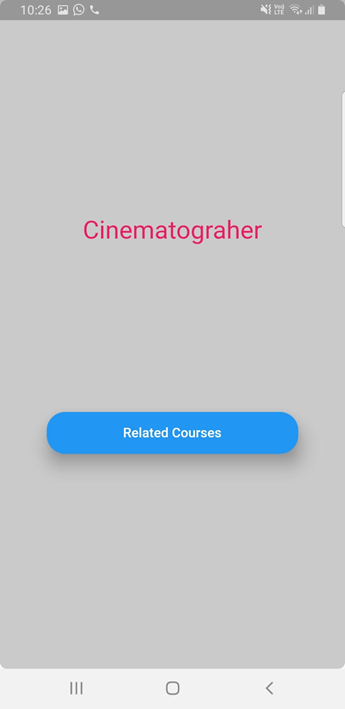
* Assessment Page-

This pageis the process of gathering information from multiple and diverse sources in order to develop a deep understanding of what students know, understand, and can do with their knowledge as a result of their educational experiences.

* Result Page-

This page consists of the result of prediction of a Career based on the anwers given by the User for respective Sections on the Assessment conducted in our App.

**6 CONCLUSION AND FUTURE ENHANCEMENTS-**

**Conclusion-**

Information about career guidance, challenges faced by both students and teachers in the career guidance process and the factors that ought to be considered to improve the process were reviewed. From the analysis carried out, the results pointed out that there are a few issues that the potential users wanted to be addressed. The result was the development of a career guidance system which includes a mobile application. The key features of the application include: understanding the different personalities, taking a personality test, subjective test, logical test viewing career possibilities, etc. The application was aimed at coming up with a solution that would bridge the gap in the inadequacy of personnel available both at the school and industry level for guidance and mentorship to the many students. System testing was performed, look and feel, ease of use, system functionality was looked into.

**Future Enhancements-**

The solutions provided in this research can still be improved or added upon as there is always room for improvement. This is attainable because, research is ongoing and technology is continuously improving the way things are done and more technological tools are being built daily that could further even more discoveries. Therefore, the following are areas that could be explored in future to improve on this solution:

i. Develop a web application module for the students that they can access even from schools for those are not allowed to have mobile phones while at school.

1. Expand the scope of the application by also offering guidance to university students as they are selecting their specialisation areas.
2. Improve the accuracy of the project and add more career possibilities to the dataset.

**REFERENCES-**

**Bureau., E. (2014).** *Guide on Life Planning Education and Career Guidance for**Secondary Schools*. School Development Division.

**Gordon, N. Virginia., Sears, J. Susan. (2011).** *Building Your Carer: A Guide to**Your Future. Pearson.*

**Jigau, M. (2007).** *Career Counselling compendium of Methods and**Techniques*. Bucharest: Institute of Educational Sciences, Euro guidance – Romania.

**Stoodnt, July (2018).**<https://medium.com/@stoodnt/6-advantages-of-career-guidance-e086cf49a741>

Logical and aptitude questions, <https://www.indiabix.com/>

**McPherson, B., & Mench, S. (2007*).*** *Students personality type and choice of**Major.* Academy of Information and Management Science Journal.