VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

An Internship Project Report
on

Hangman Game Application

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

by

Aman Verma
1RN18IS014

Under the Guidance of

Mr. Pramoda R
Assistant Professor
Department of ISE

ESTD:2001
An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru-560098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Internship work entitled ***Hangman game application*** has been successfully completed by **Aman Verma(1RN18IS014)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Mr.Pramoda R	**Dr. Suresh L**	**Dr. M K Venkatesha**
Internship Guide	Professor and HoD	Principal
Assistant Professor	Department of ISE	RNSIT
Department of ISE	RNSIT	

External Viva

Name of the Examiners	**Signature with Date**
1. _____	**1.** _____
2. _____	**2.** _____

DECLARATION

I, **AMAN VERMA [USN: 1RN18IS014],** student of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Hangman Game Application*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place : Bengaluru

Date :

AMAN VERMA

(1RN18IS014)

Abstract

In learning foreign language such as English, a learner should have motivation in order to have willingness to learn. This study is an attempt to investigate the students' motivation in learning English by using Hangman Game. And this study was to find out if the use of Hangman game motivates the students in learning English.

Hangman is an old school favorite, a word game where the goal is simply to find the missing word or words. You will be presented with a number of blank spaces representing the missing letters you need to find.

This project is implemented using flutter.

ACKNOWLEDGMENT

At the very onset I would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha,** for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all her wisdom.

We place our heartfelt thanks to **Mr.Pramoda R,** Assistant Professor , Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Akshay D R, Co-Founder & CEO, Enmaz Engineering Services Pvt.Ltd,** for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank our internship coordinator **Dr. Rajkumar R,** Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

Aman Verma

TABLE OF CONTENTS

List of Figures

List of Abbreviations

UAT	User Acceptance testing
SRS	Software Requirement Specification

1. Introduction

1.1 Background

As the name of the game suggests, the diagram is designed to look like a hanging man. A common alternative for teachers is to draw an apple tree with ten apples, erasing or crossing out the apples as the guesses are used up.

1.2 Existing System

In the existing system the users had limited number of words to guess hence, the game becomes less interesting as the user reaches a certain threshold ,The system also didn't have a Specialized leader board therefore making the game less competitive.

1.3 Proposed System

To overcome the drawbacks in the existing system, the proposed system has been developed. The proposed system makes the game more interesting as the user playing the game has deal with more and more words after reaching a certain threshold ,The system also has a specialized leader board therefore making the game more competitive.

2. System Design

2.1 Widget Tree

2.1.1 Game Screen

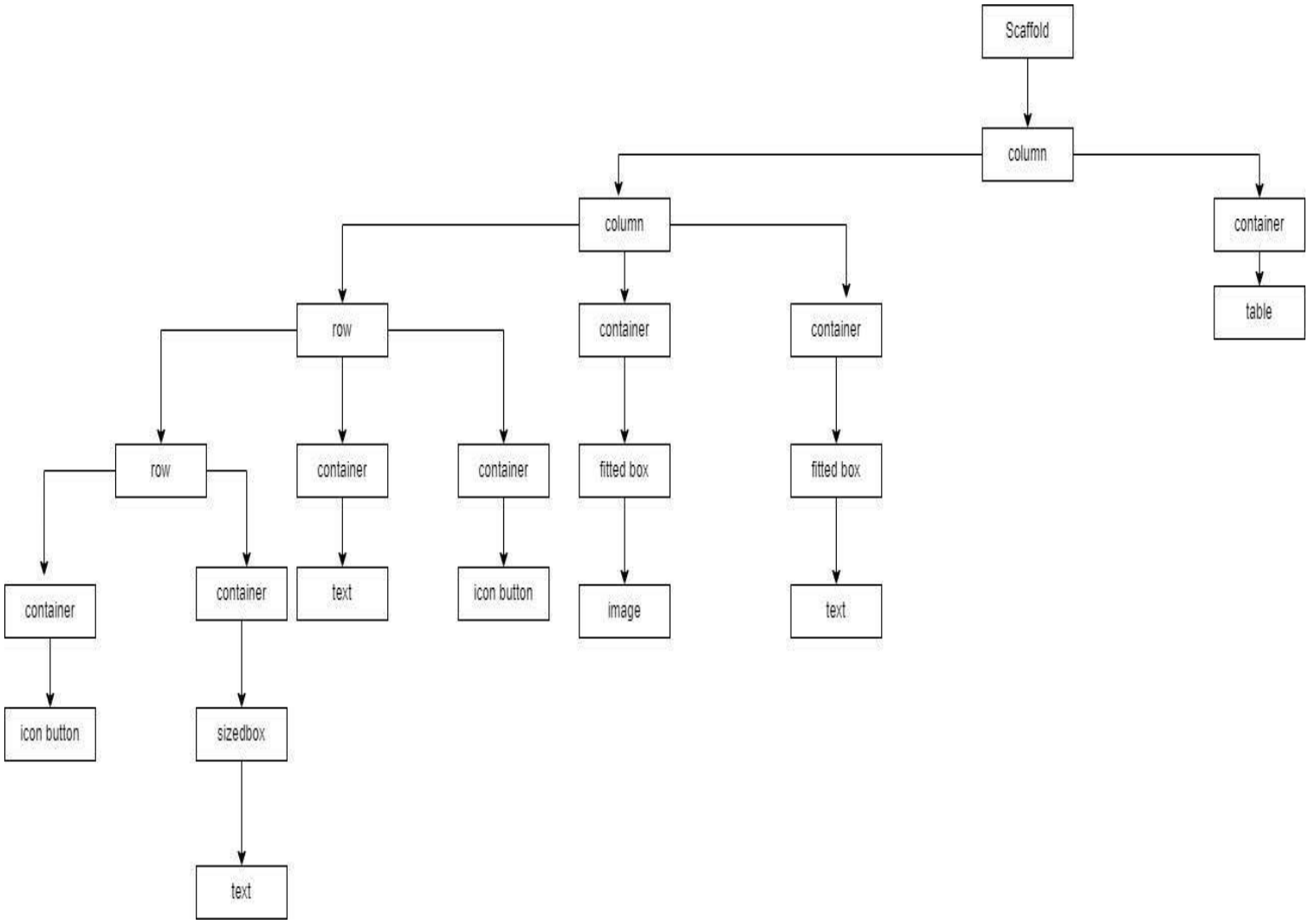

Fig 2.1.1 Widget tree for Game Screen

3. Implementation

3.1 Requirement Specifications

3.1.1 Hardware Requirements

- CPU: Pentium processor and above
- RAM: 4 GB
- HDD: 40 GB

3.1.2 Software Requirements

- **Operating System:** Windows 8 and above
- **Front-end Design:** Visual Studio Code
- **Front-end Language**: Dart

3.1.3 Flutter

Flutter is an open-source UI, software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, Web Platform and the web from a single codebase.

The major components of Flutter include:

Dart Platform Flutter

Engine

Foundation Library

Design-specific widgets

Flutter Development Tools (DevTools)

3.1.3.1 Dart Platform

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.

On Windows, macOS, and Linux Flutter runs in the Dart virtual machine, which 3 features a just-in-time execution engine. While writing and debugging an app, Flutter uses

Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful, hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

3.1.3.2 Flutter Engine

Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform Specific SDKs such as those provided by Android and 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, plug-in architecture, and a Dart runtime and compile toolchain.

3.1.3.3 Foundation Library

The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

3.1.3.4 Design Specific Widgets

The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and *Cupertino* widgets implement Apple's iOS Human Interface Guidelines.

3.2 Discussion of Code Segments

3.2.1 main.dart

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:flutter_hangman/screens/home_screen.dart';
import 'package:flutter_hangman/utilities/constants.dart';
import 'package:flutter_hangman/screens/score_screen.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    SystemChrome.setEnabledSystemUIOverlays([]);
    SystemChrome.setPreferredOrientations([
```

```
      DeviceOrientation.portraitUp,
      DeviceOrientation.portraitDown,
    ]);
    return MaterialApp(
      theme: ThemeData.dark().copyWith(
        tooltipTheme: TooltipThemeData(
          decoration: BoxDecoration(
            color: kTooltipColor,
            borderRadius: BorderRadius.circular(5.0),
          ),
          textStyle: TextStyle(
            fontWeight: FontWeight.w900,
            fontSize: 20.0,
            letterSpacing: 1.0,
            color: Colors.white,
          ),
        ),
        scaffoldBackgroundColor: Color(0xFF421b9b),
        textTheme: Theme.of(context).textTheme.apply(fontFamily:
'PatrickHand'),
      ),
      initialRoute: 'homePage',
      routes: {
        'homePage': (context) => HomeScreen(),
        'scorePage': (context) => ScoreScreen(),
      },
    );
  }
}
```

3.2.2 home_screen.dart

```
import 'package:flutter/material.dart';
import 'package:flutter/painting.dart';
import 'package:flutter_hangman/components/action_button.dart';
import 'package:flutter_hangman/utilities/hangman_words.dart';
import 'game_screen.dart';
import 'loading_screen.dart';

class HomeScreen extends StatefulWidget {
  final HangmanWords hangmanWords = HangmanWords();

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

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    double height = MediaQuery.of(context).size.height;
    widget.hangmanWords.readWords();
```

```
    return Scaffold(
      body: SafeArea(
          child: Column(
        children: <Widget>[
          Center(
            child: Container(
              margin: EdgeInsets.fromLTRB(8.0, 1.0, 8.0, 8.0),
              child: Text(
                'HANGMAN',
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 58.0,
                    fontWeight: FontWeight.w300,
                    letterSpacing: 3.0),
              ),
            ),
          ),
          Center(
            child: Container(
//              padding: EdgeInsets.all(5.0),
              child: Image.asset(
                'images/gallow.png',
                height: height * 0.49,
              ),
            ),
          ),
          SizedBox(
            height: 15.0,
          ),
          Center(
            child: IntrinsicWidth(
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.stretch,
                children: <Widget>[
                  Container(
//                    width: 155,
                    height: 64,
                    child: ActionButton(
                      buttonTitle: 'Start',
                      onPress: () {
                        Navigator.push(
                          context,
                          MaterialPageRoute(
                            builder: (context) => GameScreen(
                              hangmanObject: widget.hangmanWords,
                            ),
                          ),
                        );
                      },
                    ),
                  ),
                  SizedBox(
```

```
                    height: 18.0,
                  ),
                  Container(
//                    width: 155,
                    height: 64,
                    child: ActionButton(
                      buttonTitle: 'High Scores',
                      onPress: () {
                        Navigator.push(
                          context,
                          MaterialPageRoute(
                            builder: (context) => LoadingScreen(),
                          ),
                        );
                      },
                    ),
                  ),
                ],
              ),
            ),
          ),
        ],
      )),
    );
  }
}
```

3.2.3 game_screen.dart

```
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';
import 'package:flutter_hangman/screens/home_screen.dart';
import 'package:flutter_hangman/utilities/alphabet.dart';
import 'package:flutter_hangman/components/word_button.dart';
import 'package:flutter_hangman/utilities/constants.dart';
import 'package:flutter_hangman/utilities/hangman_words.dart';
import 'package:rflutter_alert/rflutter_alert.dart';
import 'dart:math';
import
'package:material_design_icons_flutter/material_design_icons_flutter.dart';
import 'package:flutter_hangman/utilities/score_db.dart' as score_database;
import 'package:flutter_hangman/utilities/user_scores.dart';

class GameScreen extends StatefulWidget {
  GameScreen({@required this.hangmanObject});

  final HangmanWords hangmanObject;

  @override
  _GameScreenState createState() => _GameScreenState();
```

```
}

class _GameScreenState extends State<GameScreen> {
  final database = score_database.openDB();
  int lives = 5;
  Alphabet englishAlphabet = Alphabet();
  String word;
  String hiddenWord;
  List<String> wordList = [];
  List<int> hintLetters = [];
  List<bool> buttonStatus;
  bool hintStatus;
  int hangState = 0;
  int wordCount = 0;
  bool finishedGame = false;
  bool resetGame = false;

  void newGame() {
    setState(() {
      widget.hangmanObject.resetWords();
      englishAlphabet = Alphabet();
      lives = 5;
      wordCount = 0;
      finishedGame = false;
      resetGame = false;
      initWords();
    });
  }

  Widget createButton(index) {
    return Container(
      padding: EdgeInsets.symmetric(horizontal: 3.5, vertical: 6.0),
      child: Center(
        child: WordButton(
          buttonTitle: englishAlphabet.alphabet[index].toUpperCase(),
          onPress: buttonStatus[index] ? () => wordPress(index) : null,
        ),
      ),
    );
  }

  void returnHomePage() {
    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(builder: (context) => HomeScreen()),
      ModalRoute.withName('homePage'),
    );
  }

  void initWords() {
    finishedGame = false;
    resetGame = false;
```

```
    hintStatus = true;
    hangState = 0;
    buttonStatus = List.generate(26, (index) {
      return true;
    });
    wordList = [];
    hintLetters = [];
    word = widget.hangmanObject.getWord();
//    print
    print('this is word ' + word);
    if (word.length != 0) {
      hiddenWord = widget.hangmanObject.getHiddenWord(word.length);
    } else {
      returnHomePage();
    }

    for (int i = 0; i < word.length; i++) {
      wordList.add(word[i]);
      hintLetters.add(i);
    }
  }
```

3.2.4 score_screen.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_hangman/utilities/constants.dart';
import 'package:date_format/date_format.dart';
import
'package:material_design_icons_flutter/material_design_icons_flutter.dart';

class ScoreScreen extends StatelessWidget {
  final query;

  ScoreScreen({this.query});

  List<TableRow> createRow(var query) {
    query.sort((a, b) => b.toString().compareTo(a.toString()));
    List<TableRow> rows = [];
    rows.add(
      TableRow(
        children: [
          Padding(
            padding: const EdgeInsets.only(bottom: 15.0),
            child: Center(
              child: Text(
                "Rank",
                style: kHighScoreTableHeaders,
              ),
            ),
          ),
```

```
          Padding(
            padding: const EdgeInsets.only(bottom: 15.0),
            child: Center(
              child: Text(
                "Date",
                style: kHighScoreTableHeaders,
              ),
            ),
          ),
          Padding(
            padding: const EdgeInsets.only(bottom: 15.0),
            child: Center(
              child: Text(
                "Score",
                style: kHighScoreTableHeaders,
              ),
            ),
          ),
        ],
      ),
    );
    print("${query[0]} this is query 0");
    int numOfRows = query.length;
    List<String> topRanks = ["", "", ""];
    for (var i = 0; i < numOfRows && i < 10; i++) {
      var row = query[i].toString().split(",");
      var date = row[1].split(" ")[0].split("-");
      var scoreDate = formatDate(
          DateTime(int.parse(date[0]), int.parse(date[1]),
int.parse(date[2])),
          [yy, '-', M, '-', d]);

      Widget item = TableCell(
        child: Padding(
          padding: const EdgeInsets.only(bottom: 8.0),
          child: Text(
            i < 3 ? topRanks[i] + '${i + 1}' : '${i + 1}',
            style: kHighScoreTableRowsStyle,
            textAlign: TextAlign.center,
          ),
        ),
      );
      Widget item1 = TableCell(
        child: Padding(
          padding: const EdgeInsets.only(bottom: 8.0),
          child: FittedBox(
            fit: BoxFit.scaleDown,
            child: Text(
              '$scoreDate',
              style: kHighScoreTableRowsStyle,
              textAlign: TextAlign.center,
            ),
```

```
          ),
        ),
      );
      Widget item2 = TableCell(
        child: Padding(
          padding: const EdgeInsets.only(bottom: 8.0),
          child: Text(
            '${row[0]}',
            style: kHighScoreTableRowsStyle,
            textAlign: TextAlign.center,
          ),
        ),
      );
      rows.add(
        TableRow(
          children: [item, item1, item2],
        ),
      );
    }
    return rows;
  }
```

4. Testing

4.1 Introduction

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum number of efforts. Two classes of inputs are provided to test the process

A software configuration that includes a software requirement specification, a design specification and source code.

A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

4.2 Levels of Testing

4.2.1 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

Unit testing is commonly automated, but may still be performed manually. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by-step instructional document. The unit testing is the process of testing the part of the program to verify whether the program is working correct or not. In this part the main intention is to check the each and every input which we are inserting to our file. Here the validation concepts are used to check whether the program is taking the inputs in the correct format or not.

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier. Unit test cases embody characteristics that are critical to the success of the unit.

4.2.2 Integration Testing

Integration testing is also taken as integration and testing this is the major testing process where the units are combined and tested. Its main objective is to verify whether the major parts of the program is working fine or not. This testing can be done by choosing the options in the program and by giving suitable inputs.

4.2.3 System Testing

System testing is defined as testing of a complete and fully integrated software product. This testing falls in black-box testing wherein knowledge of the inner design of the code is not a pre-requisite and is done by the testing team. System testing is done after integration testing is complete. System testing should test functional and non-functional requirements of the software.

4.2.4 Validation Testing

In this, requirements established as part of software requirements analysis are validated against the software that has been constructed. Validation testing provides final assurance that software meets all functional, behavioral and performance requirements. Validation can be defined in many ways but a simple definition is that validation succeeds when software Function in a manner that can be reasonably by the customer.

1. Validation test criteria
2. Configuration review
3. Alpha and Beta testing (conducted by end user)

4.2.5 Output Testing

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

4.2.6 User Acceptance Testing

User acceptance testing is a type of testing performed by the end user or the client to verify/accept the software application to the production environment. UAT is done in the final phase of testing.

5. Results

5.1 Home Screen

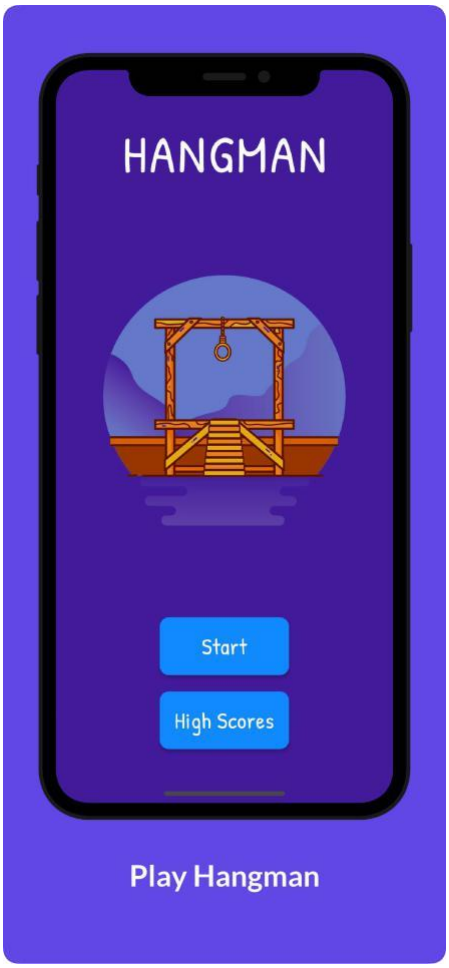

Fig. 5.1.1 Home screen

5.2 Guessing the word – initial and final stage

Fig 5.2.1 guessing the word at the initial stage

Fig 5.2.2 guessing the word at the final stage

5.3 Displaying high scores

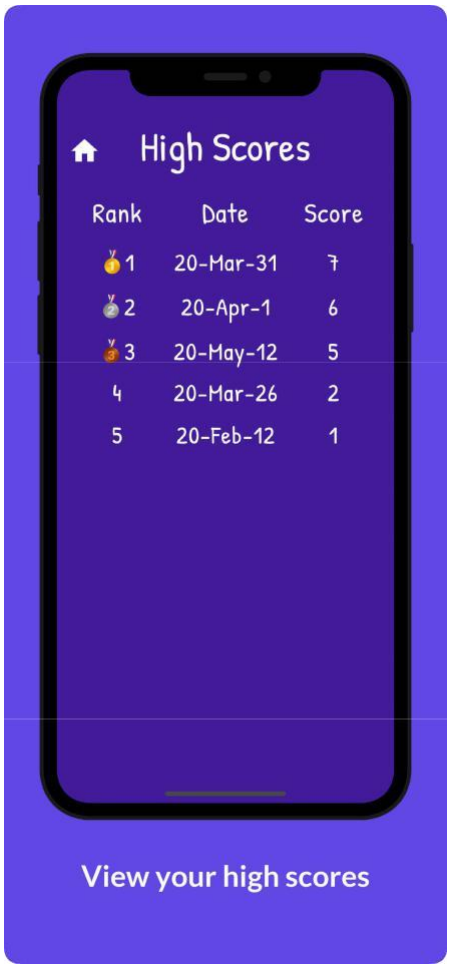

Fig 5.3 Displaying high scores

6. Conclusion and Future work

6.1 Conclusion

The main purpose of our project is to develop an application that offers new aspects of learning and improving knowledge in the educational area.

In the conclusion of this project, Hangman is a traditional game, typically played with words.

The application is user friendly and attractive thus easily playable by users of all age Groups.

The applications has a specialized leaderboard thus making the game more competitive.

The game was successfully implemented and designed so that the users playing the game successfully improve their hold on the language.

6.2 Future enhancements

This game can have varied applications in the context of word formations and puzzles. Its knowledge can be valuable to many other games like CROSSWORD PUZZLES, WHEEL OF FORTUNE, SCRABBLE.

We can also have an investigation of very popular and commonly used letters in most of the words. Make a frequency distribution in graph out of it. The underlying mathematical concepts are Data Collection and Analysis, Presentation and Interpretation which can have lot of implications in language processing and study of graphs and testing conjectures

References

Flutter : https://docs.flutter.dev/

https://stackoverflow.com/questions/tagged/flutter

The Dart Programming Language book by Gilad Bracha

https://flutter.dev/community

“Hangman Game “ : https://en.wikipedia.org/wiki/Hangman_(game)