VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI

A Project Report on

"ANAGRAMS USING PYTHON PROGRAMMING"

Submitted in the partial fulfillment for the requirements for the conferment of degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

By

Mr Mohammed Daaniyaal	USN:1BY15CS051
Mr. Abhishek B	USN: 1BY15CS005

Under the guidance of

Mrs. Mari Kirthima A

Asst. Professor,

Department of CSE, BMSIT&M.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B.M.S. INSTITUTE OF TECHNOLOGY & MANAGEMENT

Yelahanka, BENGALURU-560064

2017-2018

BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT
BANGALORE-560064

Department of Computer Science & Engineering

PBL Report of PYTHON APPLICATION PROGRAMMING 2017-2018
(6th Semester)

Batch No:	Guide Name:	Submission Date:
CSE A - 5	Mrs. Mari Kirthima A	25-05-2018

Project Title
ANAGRAMS USING PYTHON PROGRAMMING

Sl No	USN	Name
1	1BY15CS051	Mohammed Daaniyaal
2	1BY15CS005	Abhishek B

Signature of Guide

Mrs. Mari Kirthima A

Signature of HOD

Dr. THIPPESWAMY G.

ABSTRACT

An **anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.[1] For example, the word *anagram* can be rearranged into "naga ram".

The original word or phrase is known as the *subject* of the anagram.

This project is based on Anagrams i.e a simple game involving strings where the user will be shown a set of 5 or 6 letters. The task is the user should form a meaningful word with the available letters. Based on the length of the correct word the user has formed, he will be awarded so many points. Such games are also available on game portals such as Miniclip and Zapak.com. We will be trying to provide a better user interface for the users of the game along with some added features such as leader board.

Project to Program Outcomes (PO) Mapping

Project Name: Client Server Simulation Using OpenGL

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓

Program outcomes (POs):	
PO1	**Engineering knowledge:** Apply the knowledge of Mathematics, Science, Engineering fundamentals and an engineering specialization to the solution of complex engineering problems
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyse complex Engineering problems reaching substantiated conclusions using first principles of mathematics, Natural sciences and engineering sciences
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the Information to provide valid conclusions
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for Sustainable development
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings

PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering Community and with society at large, such as, being able to comprehend and write effective reports And design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a member and Leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project to Program Specific Outcomes (PSO) Mapping

Project Name: Client Server Simulation Using OpenGL

PSO1	PSO2
✓	✓

Program Specific Outcomes (PSOs):	
PSO1	Analyze the problem and identify computing requirements appropriate to its solution.
PSO2	Apply design and development principles in the construction of software systems of varying complexity.

CONTENTS

INTRODUCTION

While anagramming is certainly a recreation first, there are ways in which anagrams are put to use, and these can be more serious, or at least not quite frivolous and formless. For example, psychologists use anagram-oriented tests, often called "anagram solution tasks", to assess the implicit memory of young adults and adults alike.

Some examples of the anagrams for few words are shown:-

Good- dog, doo, god, goo

Excited- deceit, excide, edict, cited

The project demonstrates the working of a simple 2-dimensional game. It is incorporated from the online game “Anagram Magic”. It is implemented as a web application containing a front end and a back end. It would be an intellectual game where the user’s power of word formation is tested. The main focus of this game is to provide the user with an appealing user interface which would give the user a realistic game feeling. The backbone of this project is strings. The user will be provided a string containing random characters and he has to form a meaningful word out of them. Based on the length of the correct word formed, the user will be awarded points. Additional features such as leader board will be available and also features like real time competition with an other user online can be added later.

REQUIREMENTS SPECIFICATION

Functional Requirements –

- Intel core i3 or higher
- 1 GB graphics card
- Windows 7 or higher or Unix operating system
- Flash Player
- Web browser – Google Chrome / Mozilla Firefox / Internet Explorer
- Python 3.6 Interpreter

Non – Functional Requirements –

Non – Functional requirements support the functioning of the system. They have a check on the entire system as a whole. Non – functional requirements of the system are -

- **Reliability :** How reliable is the system to deliver the system its service.

- **Safety :** The ability of the system to perform without failure.

- **Availability :** The ability of the system to render its services at all times whenever requested by the user.

- **Dependability :** This is a property of a software / system that establishes a confidence about the system to the user.

- **Security :** The system is secure such that no outside attacker can intercept the system.

IMPLEMENTATION

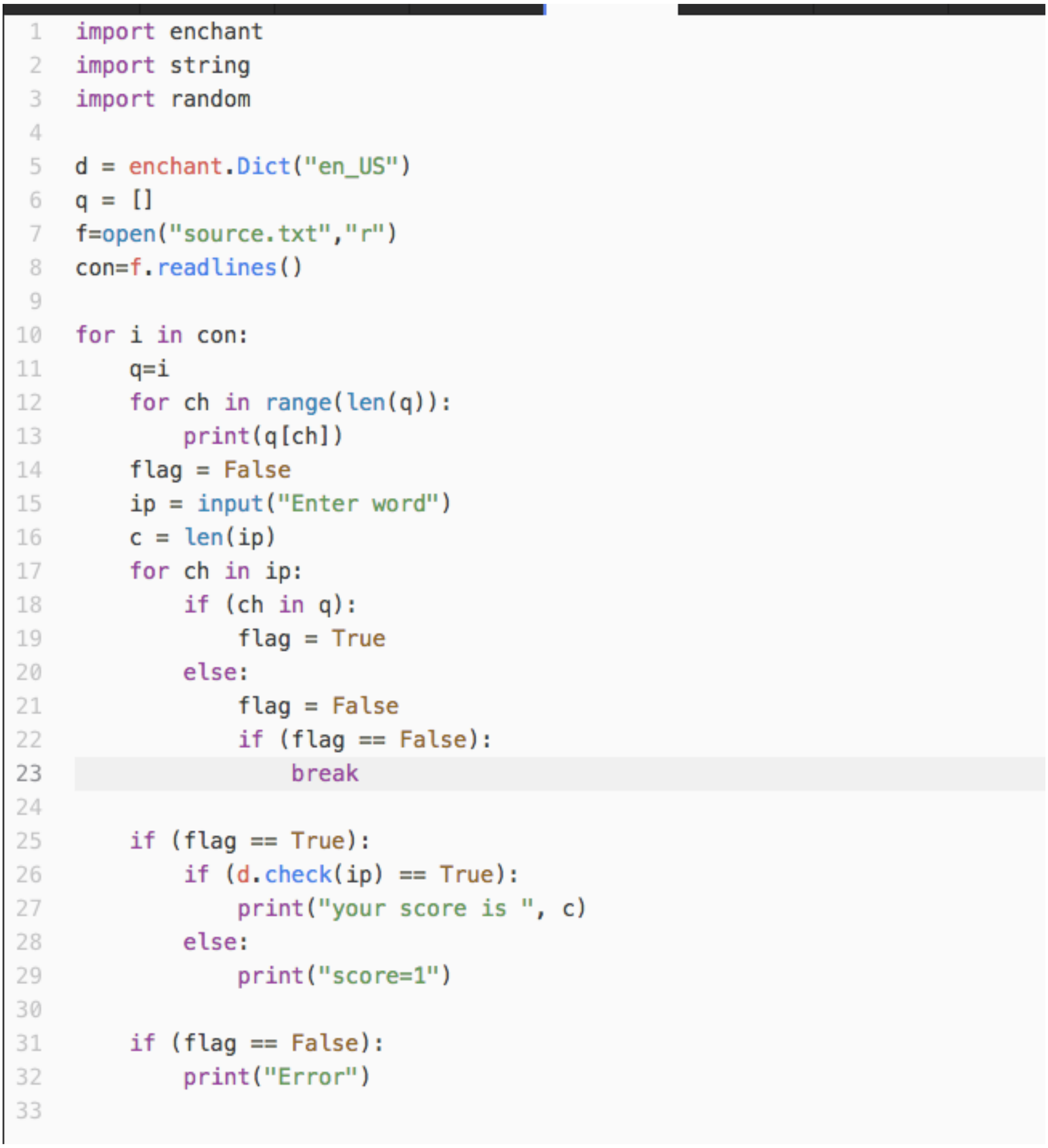
There are different ways of implementing this project—

1. **Checking off** — Our first solution to the anagram problem will check to see that each character in the first string actually occurs in the second. If it is possible to "check - off" each character, then the two strings must be anagrams. Checking off a character will be strings in Python are immutable, the first step in the process will be to convert the second string to a list. Each character from the first string can be checked against the characters in the list and if found, checked off by replacement.

2. **Sort and Compare** — Another solution to the anagram problem will make use of the fact that even same characters. So, if we begin by sorting each string alphabetically, from a to z, we will end up with the same string if the original two strings are anagrams.

3. **Count and compare** — Our final solution to the anagram problem takes advantage of the fact that any two anagrams will have the same number of a's, the same number of b's, the same number of c's, and so on. In order to decide whether two strings are anagrams, we will first count the number of times each character occurs. Since there are 26 possible characters, we can use a list of 26 counters, one for each possible character. Each time we see a particular character, we will increment the counter at that position. In the end, if the two lists of counters are identical, the strings must be anagrams.

CODE :



```python
import enchant
import string
import random

d = enchant.Dict("en_US")
q = []
f=open("source.txt","r")
con=f.readlines()

for i in con:
    q=i
    for ch in range(len(q)):
        print(q[ch])
    flag = False
    ip = input("Enter word")
    c = len(ip)
    for ch in ip:
        if (ch in q):
            flag = True
        else:
            flag = False
            if (flag == False):
                break

    if (flag == True):
        if (d.check(ip) == True):
            print("your score is ", c)
        else:
            print("score=1")

    if (flag == False):
        print("Error")
```

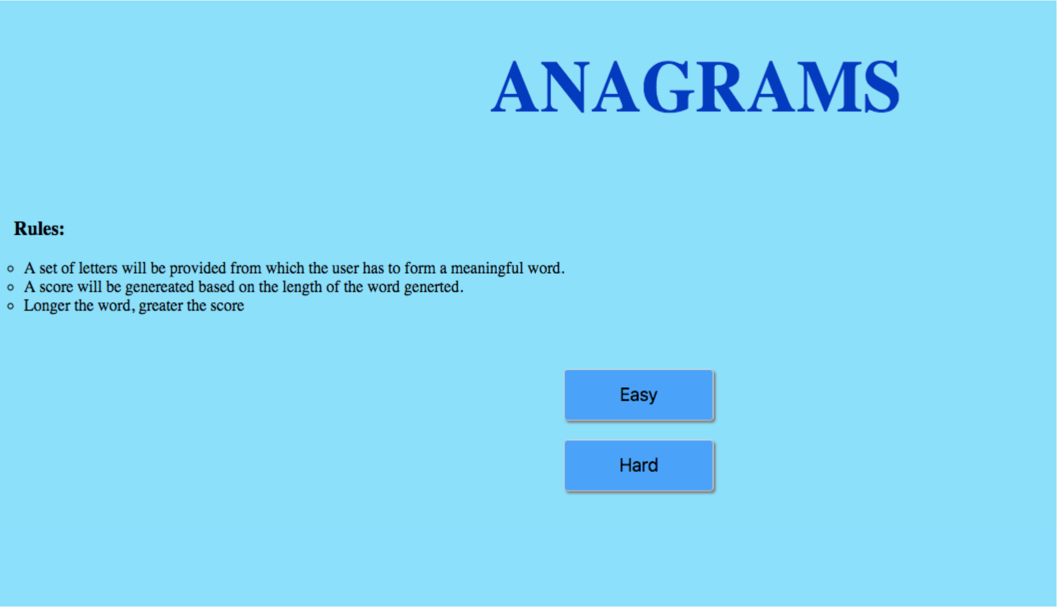
The system is implemented using Django 2.0 which is web framework for Python.

Django makes the system highly scalable and maintainable.

A python package called **pyenchant** is used to check if the word entered by the user is a valid English word.

INTERPRETATION OF RESULTS

1. **Start Page-**

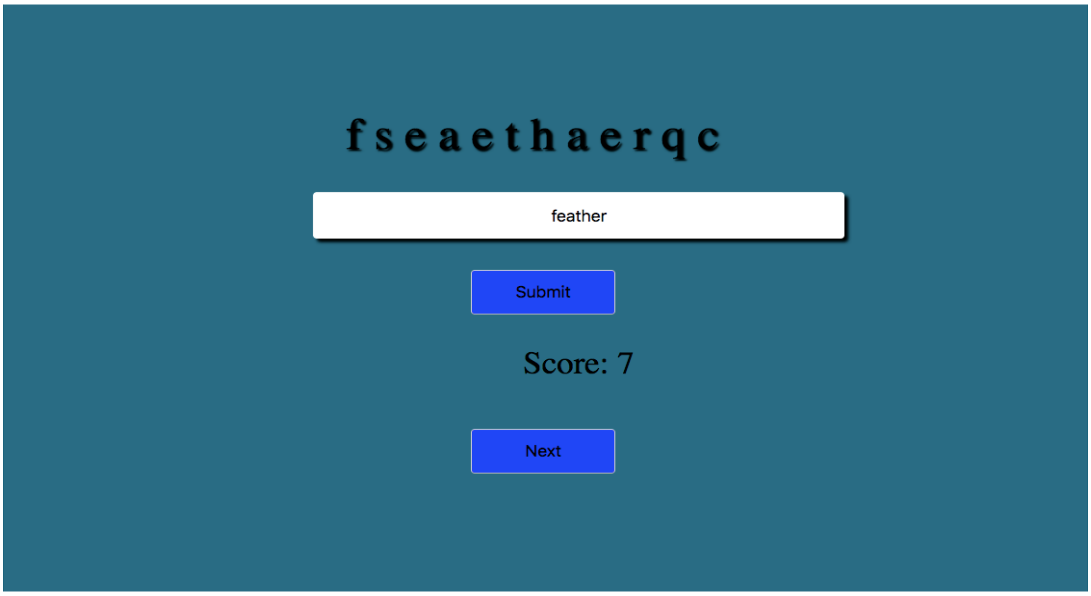


ANAGRAMS

Rules:

- A set of letters will be provided from which the user has to form a meaningful word.
- A score will be genereated based on the length of the word generted.
- Longer the word, greater the score

2. **1st word**



fseaethaerqc

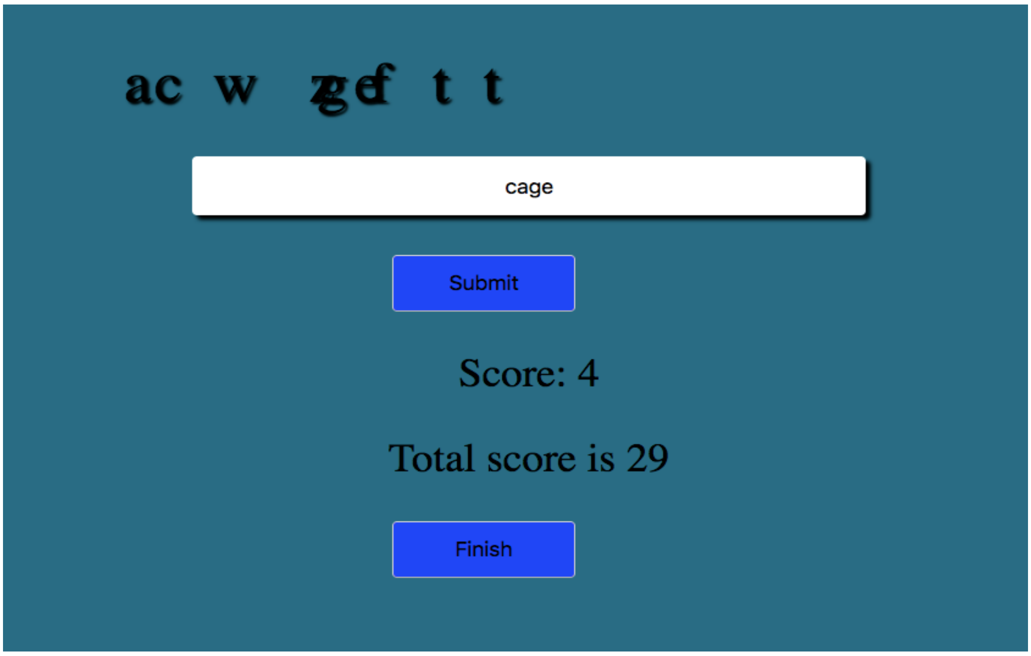
feather

Score: 7

The jumbled letters are displayed and the input is provided in the text box and submitted.

Based on the length of the correct word, the score is displayed.

3. Last word



Score: 4

Total score is 29

On coming to the last word, the total score is calculated and displayed on the screen.

On clicking the finish button, the game comes to an end.

Conclusion

The users of this system will be able to play this game to improve their vocabulary and learn English in a fun way. This improves thinking capabilities of the user. It tests the users ability to make meaningful English words from a given set of jumbled letters. The motive of this project is to build an efficient user interface for the users by using animations and colors. We therefore conclude this project by hoping that the stated motives have been achieved.

FUTURE ENHANCEMENTS

- Multi-player mode.
- Include a timer to decide the winner of the game.
- Improve User Interface (UI) / User Experience (UX)
- Improve performance of the system.

REFERENCES

- https://stackoverflow.com/
- www.sourceforge.net
- https://www.youtube.com/
- https://www.wikipedia.org/
- The Python Workbook – Ben Stephenson