BILKENT UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

CS461-ARTIFICIAL INTELLIGENCE

Term Project Report

New York Times Alternate Clue Generator

HUKAB

1) Baran Uslu
2) Kenan Şahın
3) Hasaan Ijaz
4) Mohammad Usayrim Rehman
5) Syed Abdullah Hassan

MAY 13, 2020

Introduction:

Puzzles are considered to be beneficial for developing cognitive skills, as they are known for thoroughly engaging the brain. One such puzzle is the New York Time 5 x 5 crossword puzzle, which stimulates the brain by requiring the player to guess a word or a term with the help of the provided clues. The New York Times website, however, only gives its users the privilege of being able to solve only one 5 x 5 crossword puzzle per day, which for some people, looking forward to enhance their cognitive skills, is not enough. Our CS461 Artificial Intelligence (AI) project is of great help to such people, as it requires our group HUKAB to design an artificially intelligent program which takes the New York Times 5 x 5 crossword puzzle, and replaces its clues with a brand new set of clues, making each puzzle playable more than once. The across and down positioning (geometry), as well as the answers to the puzzle remain the same, however, the set of clues will be replaced by a new one allowing the players to broaden their knowledge by learning multiple definitions of the same term, which are found by accessing a variety of definition sources across the Internet.

Problem Solving Strategy:

The purpose of our artificially intelligent program, as described, was to obtain new clues for 5 x 5 New York Times crossword puzzle using the original puzzle, its solution, and the original clues. To tackle the problem, and train HUKAB to solve the problem, we began brainstorming all possible ways to tackle the problem. Our obvious first instinct was to somehow get hold of the solutions to the crossword puzzle, and then using them to search the web for alternate definitions that after meeting certain rules would act as the new clues to the puzzle.

We came with a strategy to program HUKAB in a way that it uses the solutions to the puzzle as search words to search online dictionaries, mainly Wordnet, for new definitions of the word. While searching, we tried to analyze the following four attributes of the search word that might lead us to our new clue:

1) Definitions: Can directly act as the new clue in most cases by describing a word from a different perspective
2) Synonyms: Single words that might act as clues for the search word.
3) Antonyms: Can also be used as a hint to help guess the search word.
4) Use of the search word in a sentence: these sentences can then be used as new clues by leaving out the search word blank. This attribute is not very reliable, as more often than not these sentences contain the search word without context, and the word is not properly embedded in the sentences. As an instance, the sentence “The hunter killed the Tiger” can’t act as a clue for the search word, Tiger because when the search word is removed from the sentence, the sentence stops acting as a hint, since the phrase “The hunter killed the _____________” can fit multiple unrelated terms and still make sense, and therefore can’t be used as a hint.

Hence, the last attribute was used as the last resort while we programmed HUKAB to generate new clues for a 5 x 5 New York Times crossword puzzle.

Brief Overview of Our Design:

As instructed, we designed HUKAB in a way that it replaced the old clues of the daily 5 x 5 New York Times[1] crossword puzzle with new ones. The first step in this design included retrieving all necessary information from the New York Time website that was to be fed to our program. This included:

1) The Original Down and Across Clues.
2) Solutions to the Puzzle.
3) The Structure of the Puzzle including the numbering and the placement of the black boxes.

For the purpose of information extraction, we used the Selenium[2] library, which alleviates the extraction process by automating web applications for testing purposes. Our program, HUKAB, uses Selenium to interact with the New York Times website by clicking the buttons required to reveal the necessary information that needs to be scraped from the website.

After the essential information has been retrieved, we move on to the next step, which involves a procedure that uses the answers to the puzzle (scraped from the website) as a search term to explore the internet for alternate definitions of the word. This involves searching different webpages (Wordnet and Merriam Webster Dictionary) for alternative definitions of the search word, and then prescribing the following set of rules for the generation of new clues:

1) **The new clue generated must differ from the old clue.** This is fairly obvious because if not met defeats the whole purpose of the project. Even though, it's highly unlikely we find a definition similar to our original clue, but the rule must be includes as a precaution.
2) **If the program generates a blank/empty clue, it should be disregarded.** Fairly obvious again, as it's difficult to guess a word based on an empty string.
3) **If our clue contains the answer, the answer part must be blanked out.** This is based on the fourth property of finding new clues, and on the fact that the New York Times website never includes the answer inside the clues.
4) **The new clue must be between 20 and 150 characters long.** Just an observation that the New York Times website follows this for its 5 x 5 crossword puzzle.

When a definition meets these conditions, our program (HUKAB) takes it as the new clue. HUKAB also takes special measures for generating clues for particularly hard terms such as abbreviations, names of celebrities, and other terms that are not particularly in the dictionary.

When the New York Times website has been scraped, and new clues have been generated, the Tkinter library in Python is used to design the Graphical User Interface (GUI) that displays the puzzle grid alongside the original, and the newly generated clues.

Designing HUKAB:

Step 1-Web Scraping and Puzzle Parsing:

Owing to our familiarity with the language, we decided to design HUKAB using the Python programming language, as its automated libraries alleviate the job for us. As described before that in order to explore the web for new clues, we need to retrieve the 5 x 5 crossword puzzle itself, its solutions, and its original set of clues from the New York Times website.

For the extraction procedure, our group decided to make use of the renowned Selenium framework as it provides us the privilege of manipulating browser functionality by making use of automated programming scripts. The dire need for browser manipulation using Selenium arose because the answers to the 5 x 5 crossword puzzle were not readily available by just requesting the website using its URL, and the "Reveal Puzzle" button needed to be pressed whose listener was programmed using the JavaScript. HUKAB uses the Selenium framework to scrape the daily 5 x 5 crossword from the New York Times website by following the following steps:

1) Opens the browser, and sends a request to New York Times website.
2) The Selenium code waits until the webpage loads completely.
3) Presses the "Reveal Puzzle" button
4) The crosswords HTML code is computed and the structure, original clues, and the solution is recorded into a JSON file
5) The browser is shut.

Step 2-Designing the Graphical User Interface (GUI) using Tkinter:

After the required data had been successfully retrieved, the project required our artificially intelligent program to display the solved puzzle along with the clues using Graphical User Interface (GUI). HUKAB accomplished this by employing Tkinter[3] (a built in module in Python). The user interface we designed included a canvas for showing the solved puzzle, and labels to exhibit the down and across hints singly. In our interface, space was purposely left empty for new clues upon generating which the Graphical User Interface code was modified to display the newly generated clues below the original clues.

Step 3-Finding New Clues:

This is perhaps the most pivotal phase in the development of our artificially intelligent system, as the abstract of the project revolves around this phase that involves finding new clues to replace the old ones. Our work up till now has led to the accumulation of information that has the following characteristics

1) A solved puzzle consisting of the positioning of black boxes, the solutions of the puzzle, and the box numberings.
2) The original down and across clues corresponding to the number on the puzzle

Using this as a stepping stone, we isolated this phase of our design of HUKAB from the steps of web scraping and development of a graphical user interface. Our purpose of doing so was to develop this pivotal phase of the task in isolation with the rest of the project use its essence as "blackbox" to revamp the consistency of the design skeleton.

Resources used in the Generation of Clues:

Wordnet:

While investigating which website contained definitions of a comprehensive list of words and terms, Wordnet[4] topped our preference, as using it substantially abridged our dependency on other websites. To begin the clue finding phase, we designed a function that takes in a solution of the puzzle as a search word and looks it up in the dictionary database of Wordnet. Wordnet was developed by Princeton University in the United States, and even though it grants access to the source files so that you can access the webpage offline, we still deemed it more efficient to make use of the Natural Language Toolkit (NLTK) [5]library which consists of the entire Wordnet database as it known widely and highly in the domain of Natural Language Processing (NLP) which is the backbone of designing artificially intelligent programs that teach themselves to operate in a language known to humans. The tutorial provided made it easier for us to make a way through familiarizing ourselves with the essential portions of the operating body. Another reason as to why Wordnet played such an important role in our project was that it provided us with all the proposed attributes (synonyms, antonyms, and sample phrases) of the specified search words. The Natural Language Toolkit (NLTK) has a Wordnet library that contains all of the four proposed characteristics in its directory. All we have to do is feed our artificially intelligent program with the search word, and it will return all the required characteristics as a type usable in Python (dictionary). Similarly, if Wordnet fails to entertain a search word, we receive an empty list which forces us to check in our next resource: The Merriam Webster Dictionary

Merriam Webster Dictionary:

After failing to find our clue on Wordnet we move on to the Merriam Webster Dictionary[6] where we need a code to explore the contents of the dictionary which is done by using the website's API. If we have the website's API key, we send a ping with the search word attached in the uniform resource locator. If all goes well, the dictionary returns the data relevant to the search word in a Python readable format, which when properly parsed can give us a clue that replaces the original clue by meeting the proposed search word characteristics.

After completing the clue generation phase, an essential function was written and created that generates a new clue for each word retrieved from the New York Times website. The function created is fed with the extracted puzzle information, and sends back the new hints in the same format as the original ones (across and down). When the code for the function was written, we called the function in the graphical user interface module so that the new clues are now displayed on the user interface.

Conclusion:

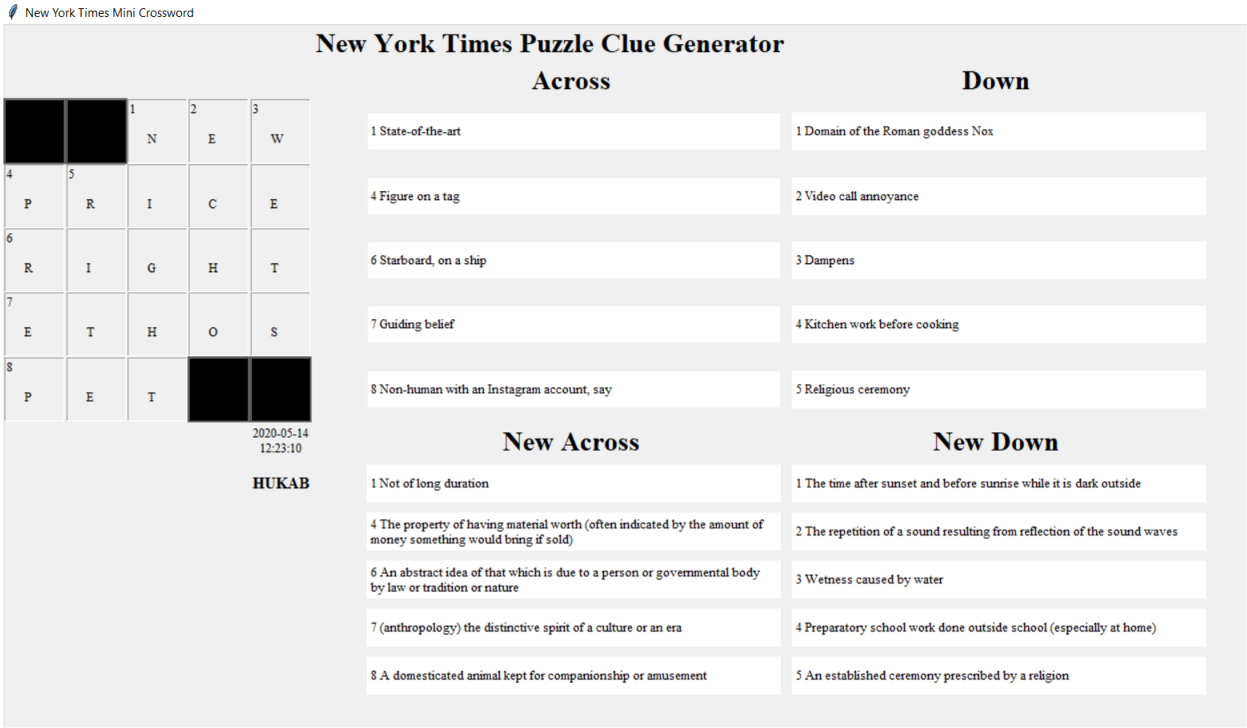
Our project worked efficiently for most test runs, and the error rate was within the range of experimental accuracy proving that our artificially intelligent system was well designed, as the quality of new clues generated was similar to those on New York Times website, and the AI worked accurately most of the time, giving it both precision and accuracy. The most common source of error was that in some cases when our program accessed the Merriam Webster dictionary, its website denied the connection resulting in the generation of no new clue.

This project reports work done in partial fulfillment of the requirements for CS 461 -- Artificial Intelligence. The software is, to a large extent, original (with borrowed code clearly identified) and was written solely by members of the project group. - HUKAB

(Word Count: 2052)

Appendix:

1-Test Runs



New York Times Mini Crossword

New York Times Puzzle Clue Generator

■	■	1 N	2 E	3 W
4 P	5 R	I	C	E
6 R	I	G	H	T
7 E	T	H	O	S
8 P	E	T	■	■

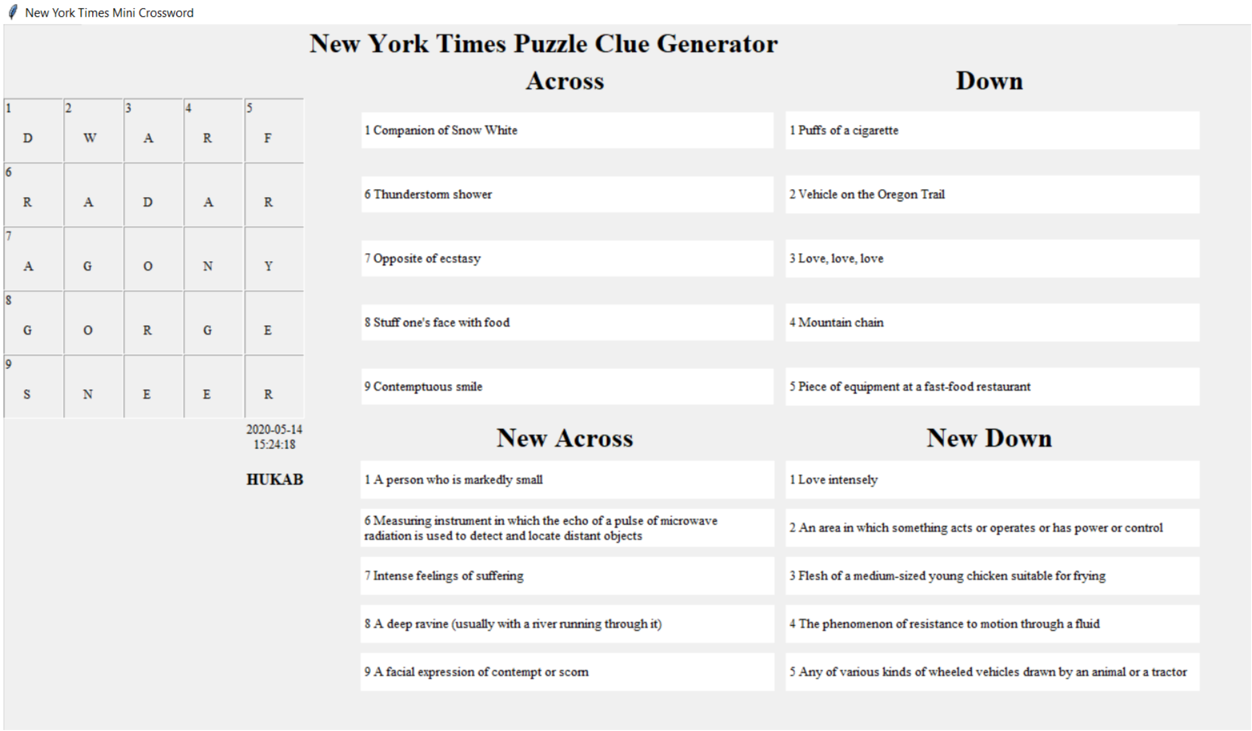
2020-05-14
12:23:10

HUKAB

Across	Down
1 State-of-the-art	1 Domain of the Roman goddess Nox
4 Figure on a tag	2 Video call annoyance
6 Starboard, on a ship	3 Dampens
7 Guiding belief	4 Kitchen work before cooking
8 Non-human with an Instagram account, say	5 Religious ceremony

New Across	New Down
1 Not of long duration	1 The time after sunset and before sunrise while it is dark outside
4 The property of having material worth (often indicated by the amount of money something would bring if sold)	2 The repetition of a sound resulting from reflection of the sound waves
6 An abstract idea of that which is due to a person or governmental body by law or tradition or nature	3 Wetness caused by water
7 (anthropology) the distinctive spirit of a culture or an era	4 Preparatory school work done outside school (especially at home)
8 A domesticated animal kept for companionship or amusement	5 An established ceremony prescribed by a religion

Test 1



New York Times Mini Crossword

New York Times Puzzle Clue Generator

1 D	2 W	3 A	4 R	5 F
6 R	A	D	A	R
7 A	G	O	N	Y
8 G	O	R	G	E
9 S	N	E	E	R

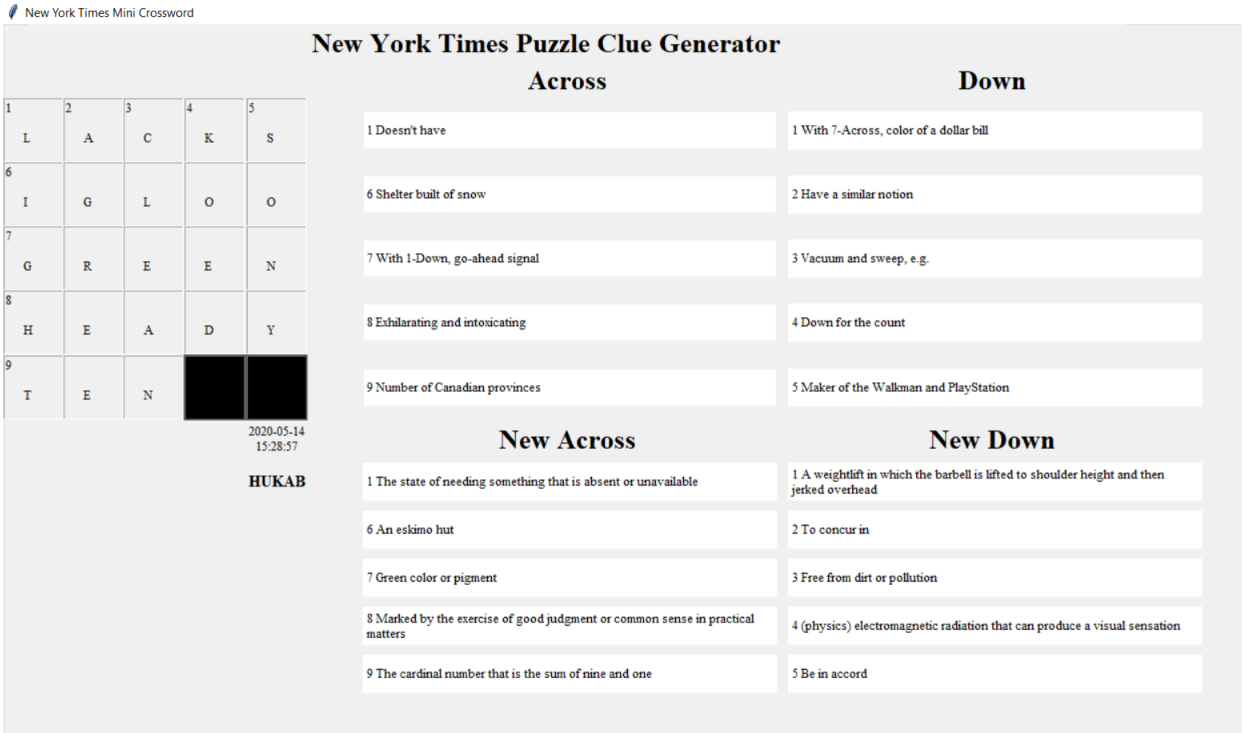
2020-05-14
15:24:18

HUKAB

Across	Down
1 Companion of Snow White	1 Puffs of a cigarette
6 Thunderstorm shower	2 Vehicle on the Oregon Trail
7 Opposite of ecstasy	3 Love, love, love
8 Stuff one's face with food	4 Mountain chain
9 Contemptuous smile	5 Piece of equipment at a fast-food restaurant

New Across	New Down
1 A person who is markedly small	1 Love intensely
6 Measuring instrument in which the echo of a pulse of microwave radiation is used to detect and locate distant objects	2 An area in which something acts or operates or has power or control
7 Intense feelings of suffering	3 Flesh of a medium-sized young chicken suitable for frying
8 A deep ravine (usually with a river running through it)	4 The phenomenon of resistance to motion through a fluid
9 A facial expression of contempt or scorn	5 Any of various kinds of wheeled vehicles drawn by an animal or a tractor

Test 2



New York Times Mini Crossword

New York Times Puzzle Clue Generator

1 L	2 A	3 C	4 K	5 S
6 I	G	L	O	O
7 G	R	E	E	N
8 H	E	A	D	Y
9 T	E	N	■	■

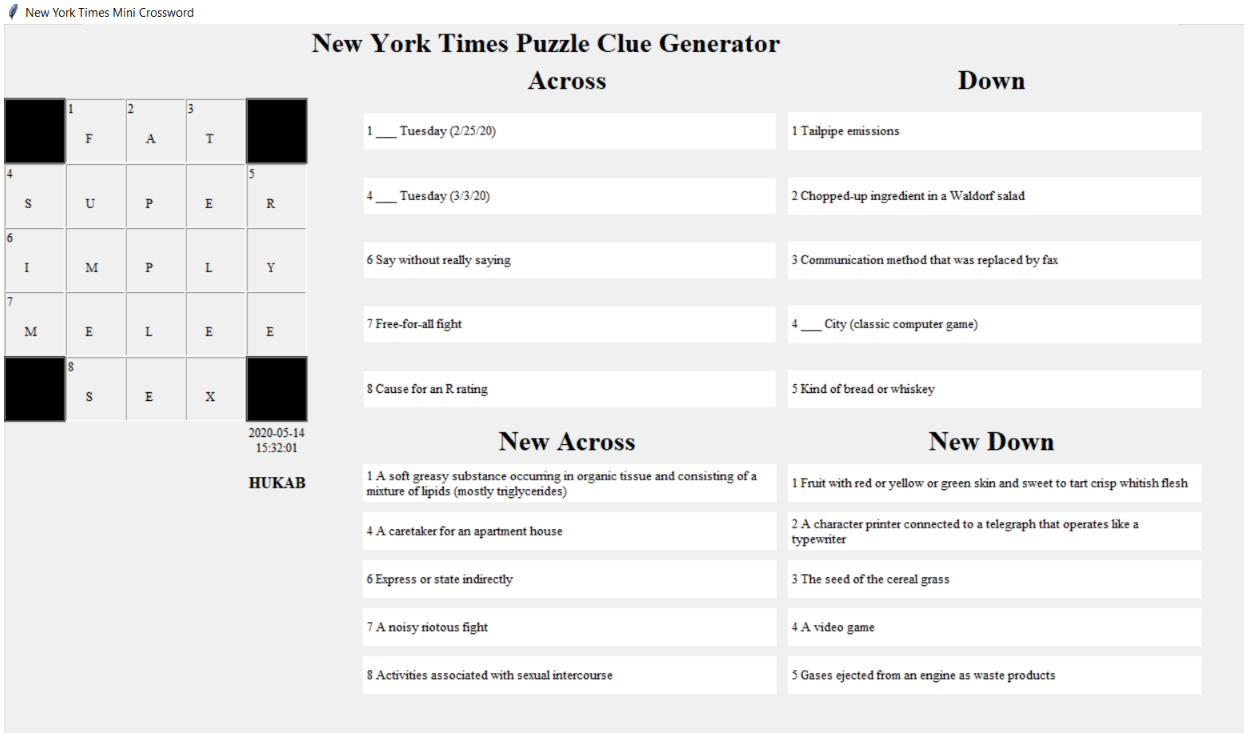
2020-05-14
15:28:57

HUKAB

Across	Down
1 Doesn't have	1 With 7-Across, color of a dollar bill
6 Shelter built of snow	2 Have a similar notion
7 With 1-Down, go-ahead signal	3 Vacuum and sweep, e.g.
8 Exhilarating and intoxicating	4 Down for the count
9 Number of Canadian provinces	5 Maker of the Walkman and PlayStation

New Across	New Down
1 The state of needing something that is absent or unavailable	1 A weightlift in which the barbell is lifted to shoulder height and then jerked overhead
6 An eskimo hut	2 To concur in
7 Green color or pigment	3 Free from dirt or pollution
8 Marked by the exercise of good judgment or common sense in practical matters	4 (physics) electromagnetic radiation that can produce a visual sensation
9 The cardinal number that is the sum of nine and one	5 Be in accord

Test 3



New York Times Mini Crossword

New York Times Puzzle Clue Generator

■	1 F	2 A	3 T	■
4 S	U	P	E	5 R
6 I	M	P	L	Y
7 M	E	L	E	E
■	8 S	E	X	■

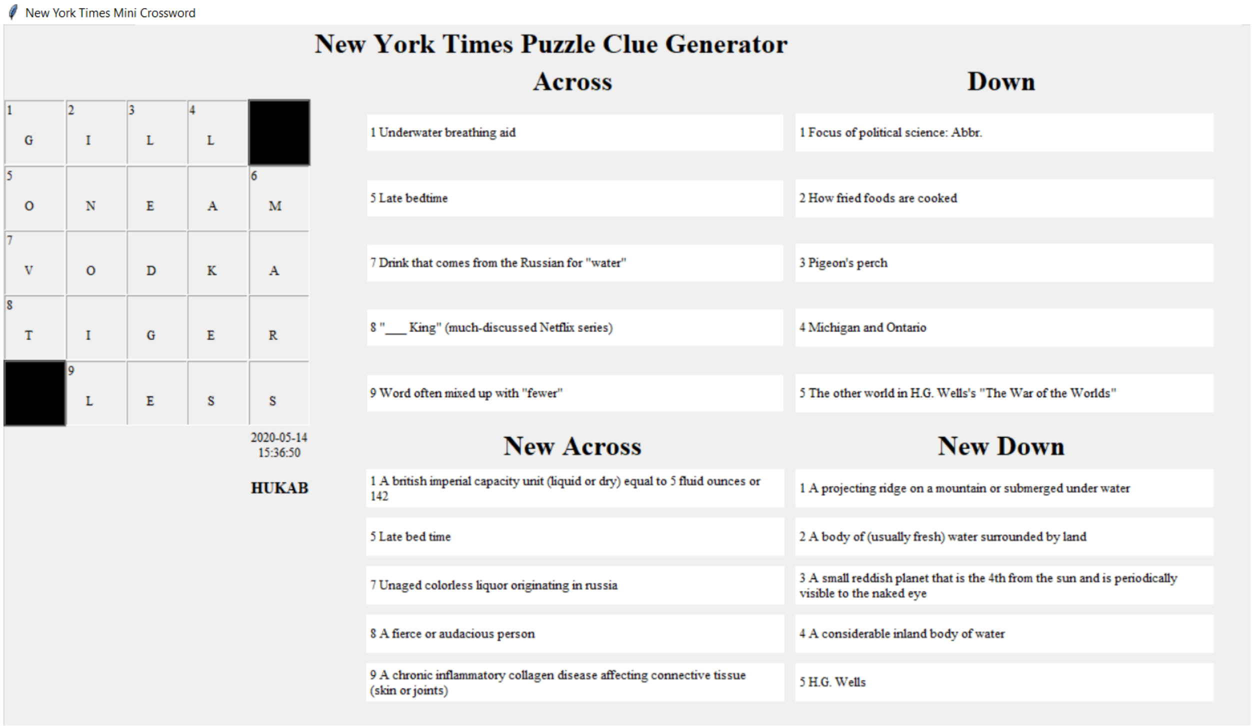
2020-05-14
15:32:01

HUKAB

Across	Down
1 ___ Tuesday (2/25/20)	1 Tailpipe emissions
4 ___ Tuesday (3/3/20)	2 Chopped-up ingredient in a Waldorf salad
6 Say without really saying	3 Communication method that was replaced by fax
7 Free-for-all fight	4 ___ City (classic computer game)
8 Cause for an R rating	5 Kind of bread or whiskey

New Across	New Down
1 A soft greasy substance occurring in organic tissue and consisting of a mixture of lipids (mostly triglycerides)	1 Fruit with red or yellow or green skin and sweet to tart crisp whitish flesh
4 A caretaker for an apartment house	2 A character printer connected to a telegraph that operates like a typewriter
6 Express or state indirectly	3 The seed of the cereal grass
7 A noisy riotous fight	4 A video game
8 Activities associated with sexual intercourse	5 Gases ejected from an engine as waste products

Test 4



New York Times Mini Crossword

New York Times Puzzle Clue Generator

1 G	2 I	3 L	4 L	■
5 O	N	E	A	6 M
7 V	O	D	K	A
8 T	I	G	E	R
■	9 L	E	S	S

2020-05-14
15:36:50

HUKAB

Across

1 Underwater breathing aid

5 Late bedtime

7 Drink that comes from the Russian for "water"

8 "___ King" (much-discussed Netflix series)

9 Word often mixed up with "fewer"

Down

1 Focus of political science: Abbr.

2 How fried foods are cooked

3 Pigeon's perch

4 Michigan and Ontario

5 The other world in H.G. Wells's "The War of the Worlds"

New Across

1 A british imperial capacity unit (liquid or dry) equal to 5 fluid ounces or 142

5 Late bed time

7 Unaged colorless liquor originating in russia

8 A fierce or audacious person

9 A chronic inflammatory collagen disease affecting connective tissue (skin or joints)

New Down

1 A projecting ridge on a mountain or submerged under water

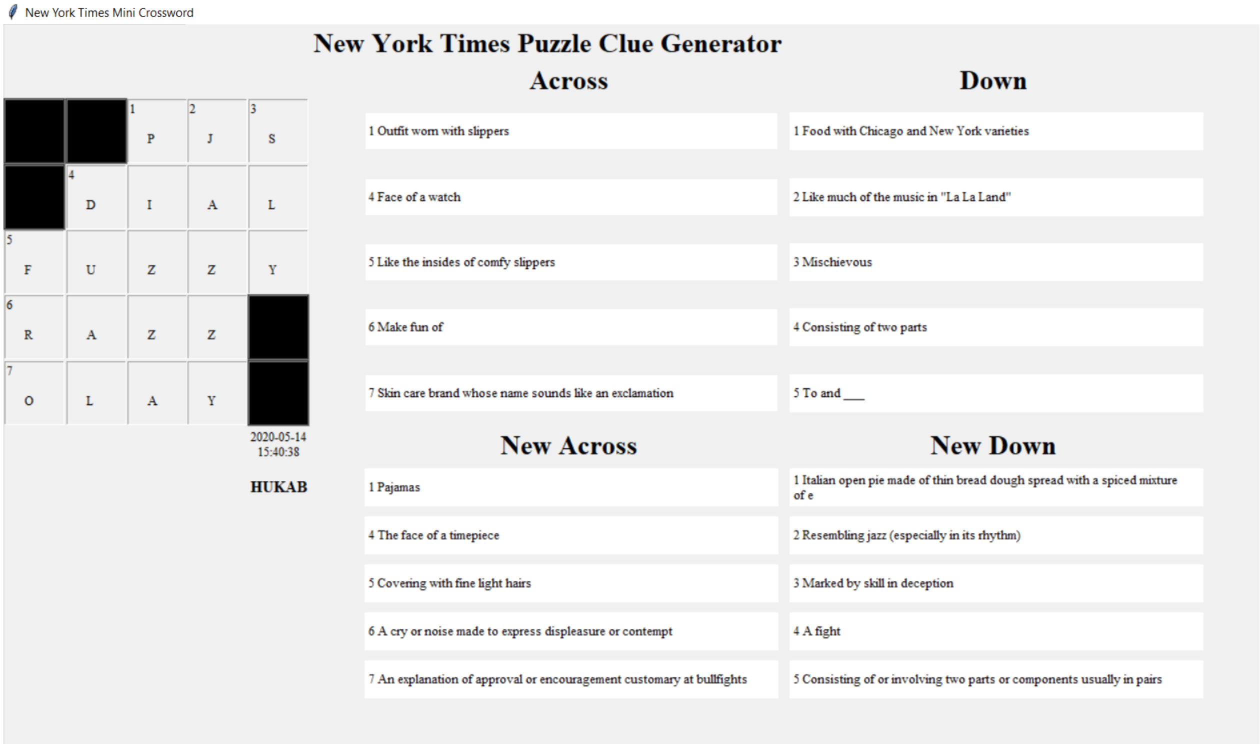
2 A body of (usually fresh) water surrounded by land

3 A small reddish planet that is the 4th from the sun and is periodically visible to the naked eye

4 A considerable inland body of water

5 H.G. Wells

Test 5



New York Times Mini Crossword

New York Times Puzzle Clue Generator

■	■	1 P	2 J	3 S
■	4 D	I	A	L
5 F	U	Z	Z	Y
6 R	A	Z	Z	■
7 O	L	A	Y	■

2020-05-14
15:40:38

HUKAB

Across

1 Outfit worn with slippers

4 Face of a watch

5 Like the insides of comfy slippers

6 Make fun of

7 Skin care brand whose name sounds like an exclamation

Down

1 Food with Chicago and New York varieties

2 Like much of the music in "La La Land"

3 Mischievous

4 Consisting of two parts

5 To and ___

New Across

1 Pajamas

4 The face of a timepiece

5 Covering with fine light hairs

6 A cry or noise made to express displeasure or contempt

7 An explanation of approval or encouragement customary at bullfights

New Down

1 Italian open pie made of thin bread dough spread with a spiced mixture of e

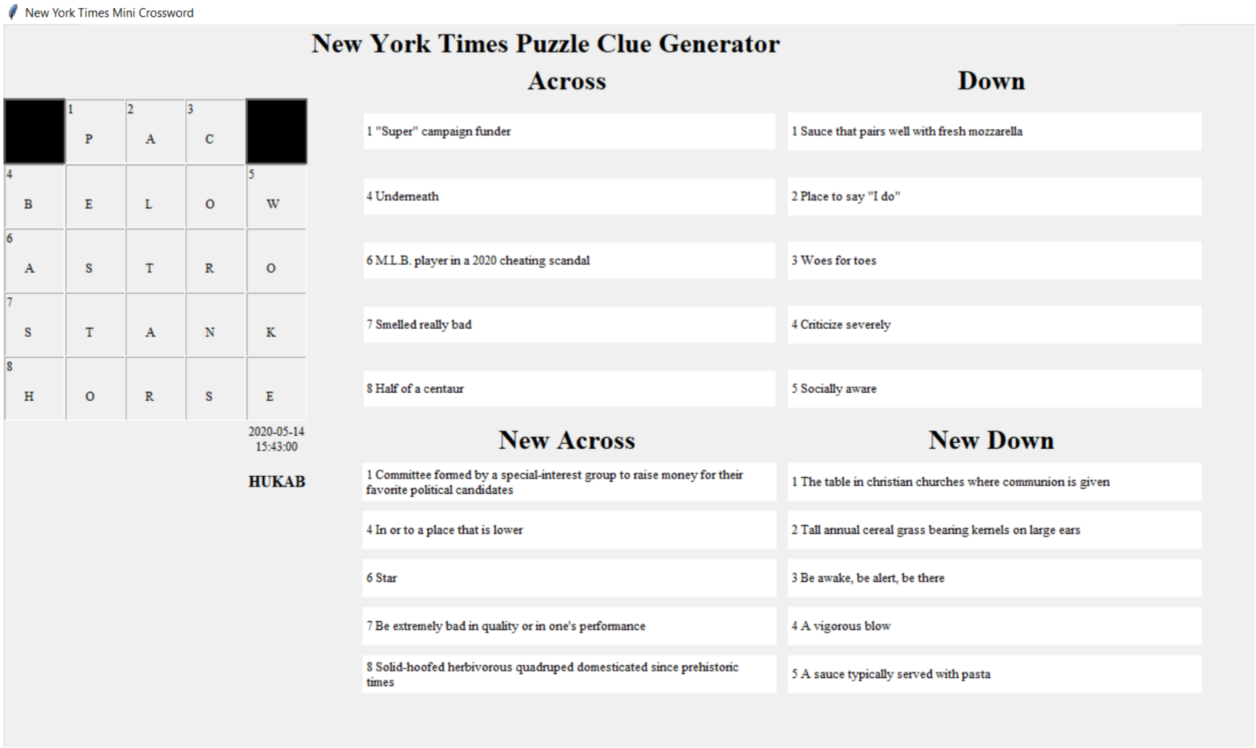
2 Resembling jazz (especially in its rhythm)

3 Marked by skill in deception

4 A fight

5 Consisting of or involving two parts or components usually in pairs

Test 6



New York Times Puzzle Clue Generator

■	1 P	2 A	3 C	■
4 B	E	L	O	5 W
6 A	S	T	R	O
7 S	T	A	N	K
8 H	O	R	S	E

2020-05-14
15:43:00

HUKAB

Across

1 "Super" campaign funder

4 Underneath

6 M.L.B. player in a 2020 cheating scandal

7 Smelled really bad

8 Half of a centaur

Down

1 Sauce that pairs well with fresh mozzarella

2 Place to say "I do"

3 Woes for toes

4 Criticize severely

5 Socially aware

New Across

1 Committee formed by a special-interest group to raise money for their favorite political candidates

4 In or to a place that is lower

6 Star

7 Be extremely bad in quality or in one's performance

8 Solid-hoofed herbivorous quadruped domesticated since prehistoric times

New Down

1 The table in christian churches where communion is given

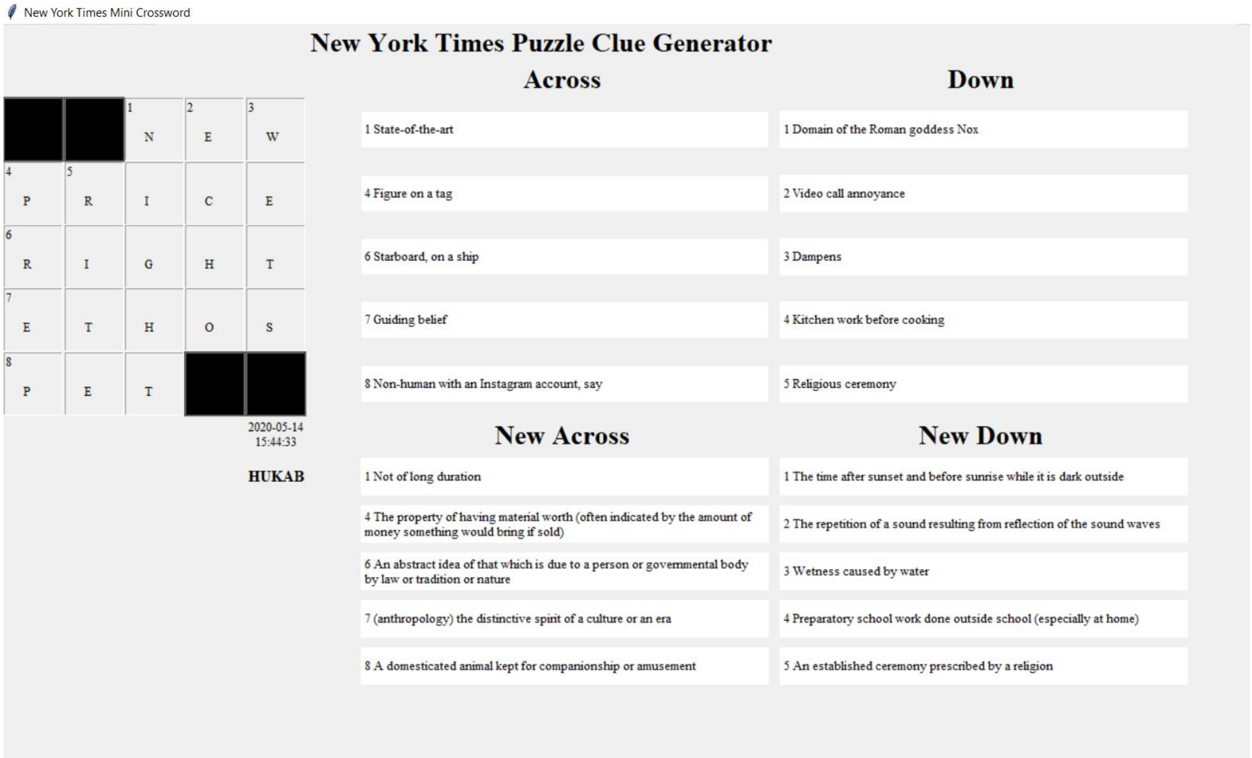
2 Tall annual cereal grass bearing kernels on large ears

3 Be awake, be alert, be there

4 A vigorous blow

5 A sauce typically served with pasta

Test 7



New York Times Puzzle Clue Generator

■	■	1 N	2 E	3 W
4 P	5 R	I	C	E
6 R	I	G	H	T
7 E	T	H	O	S
8 P	E	T	■	■

2020-05-14
15:44:33

HUKAB

Across

1 State-of-the-art

4 Figure on a tag

6 Starboard, on a ship

7 Guiding belief

8 Non-human with an Instagram account, say

Down

1 Domain of the Roman goddess Nox

2 Video call annoyance

3 Dampens

4 Kitchen work before cooking

5 Religious ceremony

New Across

1 Not of long duration

4 The property of having material worth (often indicated by the amount of money something would bring if sold)

6 An abstract idea of that which is due to a person or governmental body by law or tradition or nature

7 (anthropology) the distinctive spirit of a culture or an era

8 A domesticated animal kept for companionship or amusement

New Down

1 The time after sunset and before sunrise while it is dark outside

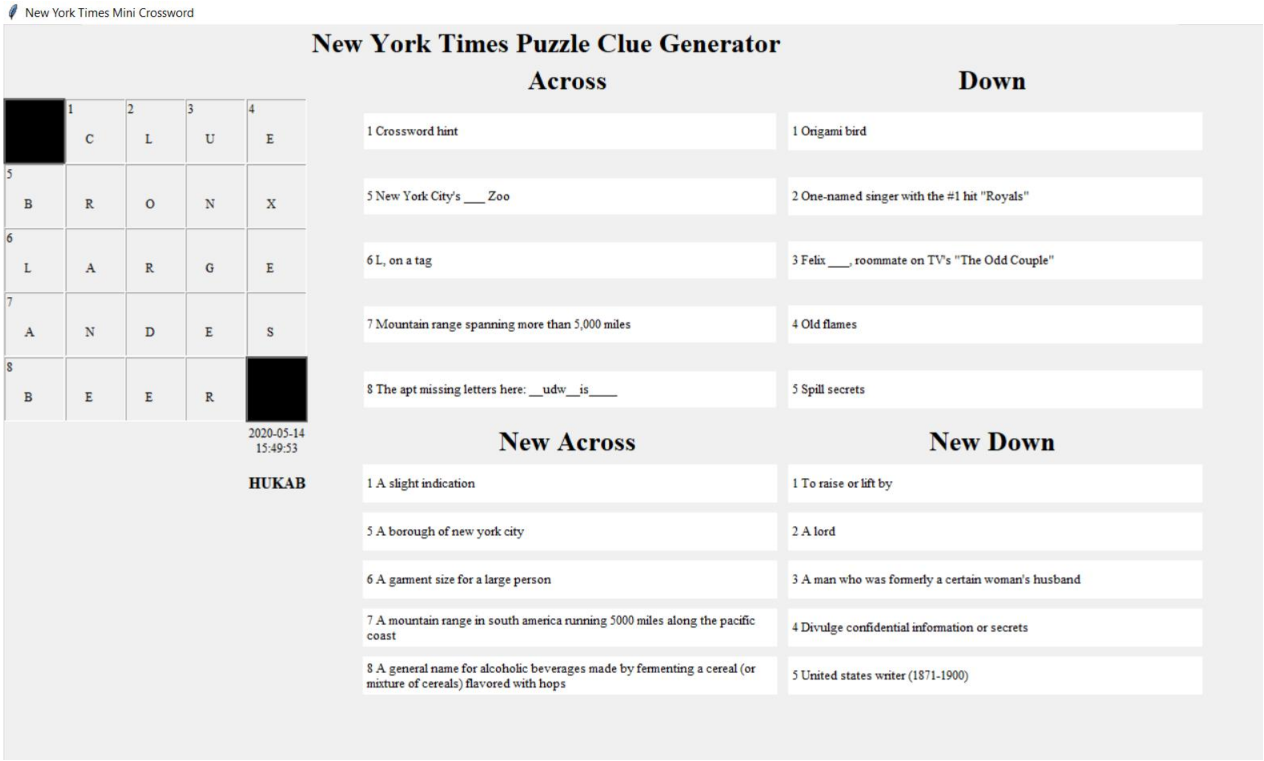
2 The repetition of a sound resulting from reflection of the sound waves

3 Wetness caused by water

4 Preparatory school work done outside school (especially at home)

5 An established ceremony prescribed by a religion

Test 8



New York Times Mini Crossword

New York Times Puzzle Clue Generator

■	1 C	2 L	3 U	4 E
5 B	R	O	N	X
6 L	A	R	G	E
7 A	N	D	E	S
8 B	E	E	R	■

2020-05-14
15:49:53

HUKAB

Across

1 Crossword hint

5 New York City's ___ Zoo

6 L, on a tag

7 Mountain range spanning more than 5,000 miles

8 The apt missing letters here: __udw__is____

Down

1 Origami bird

2 One-named singer with the #1 hit "Royals"

3 Felix ___, roommate on TV's "The Odd Couple"

4 Old flames

5 Spill secrets

New Across

1 A slight indication

5 A borough of new york city

6 A garment size for a large person

7 A mountain range in south america running 5000 miles along the pacific coast

8 A general name for alcoholic beverages made by fermenting a cereal (or mixture of cereals) flavored with hops

New Down

1 To raise or lift by

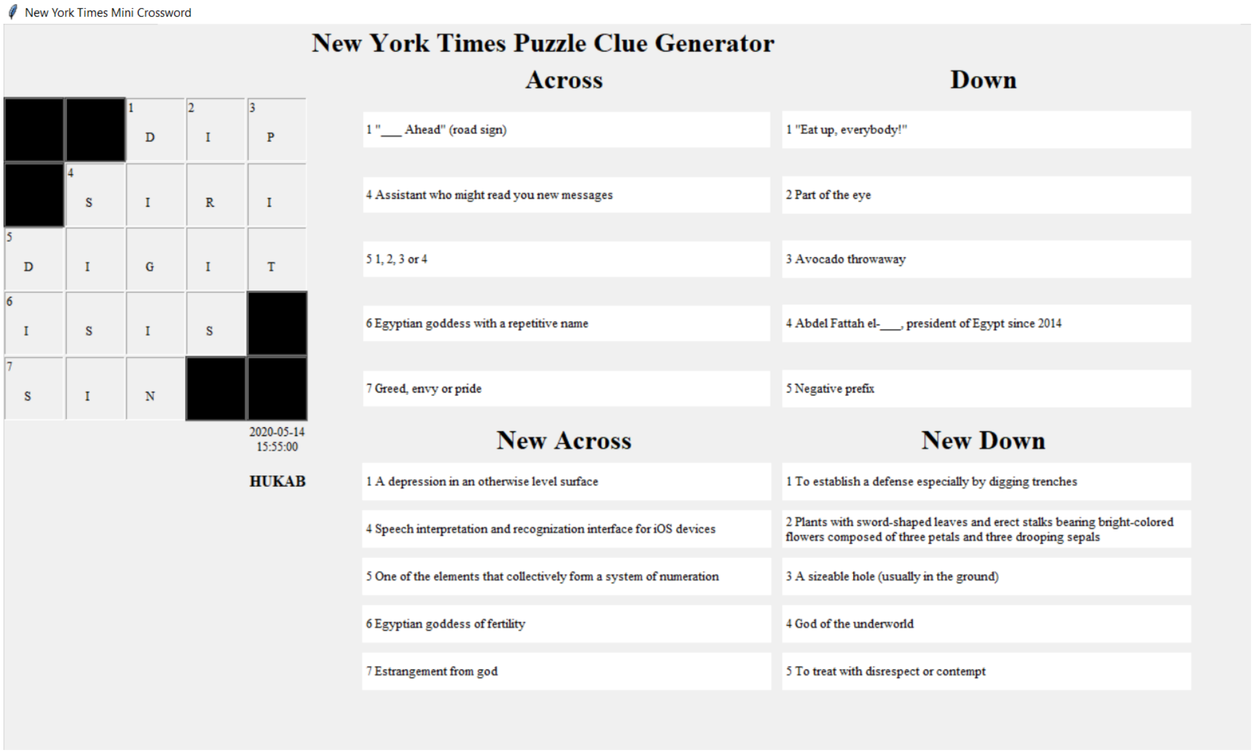
2 A lord

3 A man who was formerly a certain woman's husband

4 Divulge confidential information or secrets

5 United states writer (1871-1900)

Test 9



New York Times Mini Crossword

New York Times Puzzle Clue Generator

■	■	1 D	2 I	3 P
■	4 S	I	R	I
5 D	I	G	I	T
6 I	S	I	S	■
7 S	I	N	■	■

2020-05-14
15:55:00

HUKAB

Across

1 "___ Ahead" (road sign)

4 Assistant who might read you new messages

5 1, 2, 3 or 4

6 Egyptian goddess with a repetitive name

7 Greed, envy or pride

Down

1 "Eat up, everybody!"

2 Part of the eye

3 Avocado throwaway

4 Abdel Fattah el-___, president of Egypt since 2014

5 Negative prefix

New Across

1 A depression in an otherwise level surface

4 Speech interpretation and recognization interface for iOS devices

5 One of the elements that collectively form a system of numeration

6 Egyptian goddess of fertility

7 Estrangement from god

New Down

1 To establish a defense especially by digging trenches

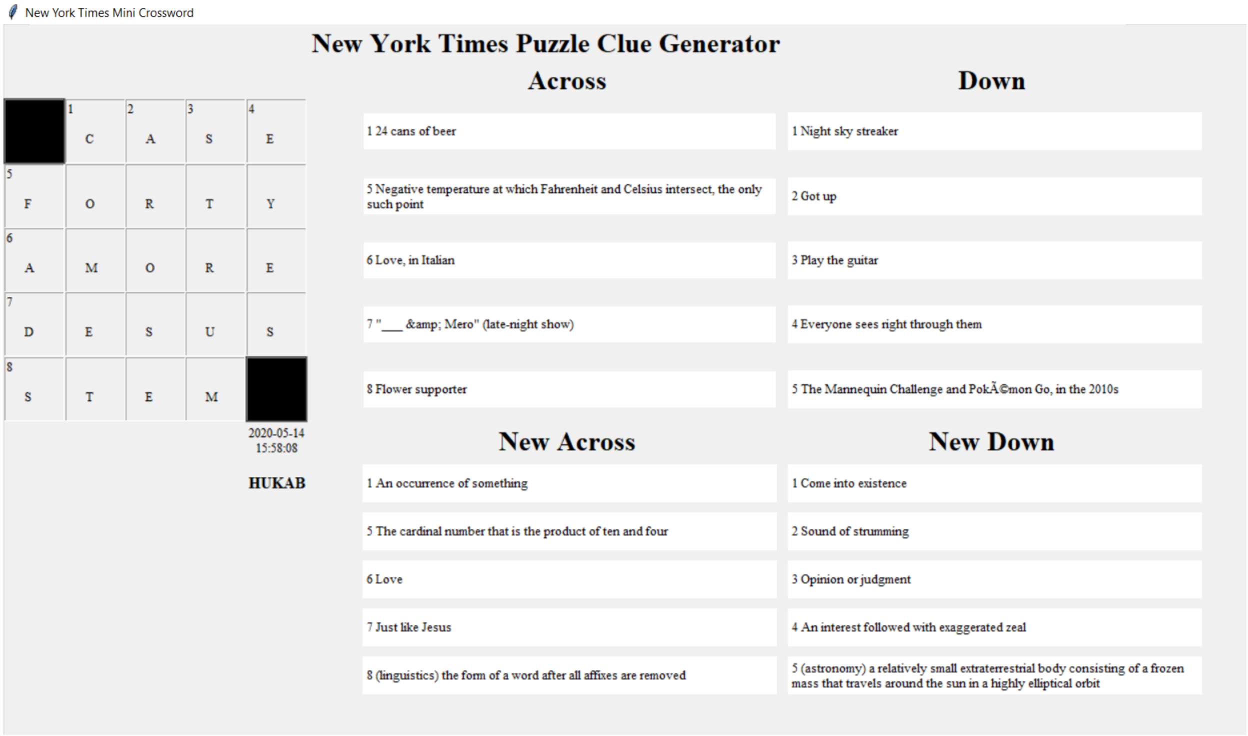
2 Plants with sword-shaped leaves and erect stalks bearing bright-colored flowers composed of three petals and three drooping sepals

3 A sizeable hole (usually in the ground)

4 God of the underworld

5 To treat with disrespect or contempt

Test 10



New York Times Mini Crossword

New York Times Puzzle Clue Generator

■	1 C	2 A	3 S	4 E
5 F	O	R	T	Y
6 A	M	O	R	E
7 D	E	S	U	S
8 S	T	E	M	■

2020-05-14 15:58:08

HUKAB

Across

1 24 cans of beer

5 Negative temperature at which Fahrenheit and Celsius intersect, the only such point

6 Love, in Italian

7 "___ &amp; Mero" (late-night show)

8 Flower supporter

Down

1 Night sky streaker

2 Got up

3 Play the guitar

4 Everyone sees right through them

5 The Mannequin Challenge and PokÃ©mon Go, in the 2010s

New Across

1 An occurrence of something

5 The cardinal number that is the product of ten and four

6 Love

7 Just like Jesus

8 (linguistics) the form of a word after all affixes are removed

New Down

1 Come into existence

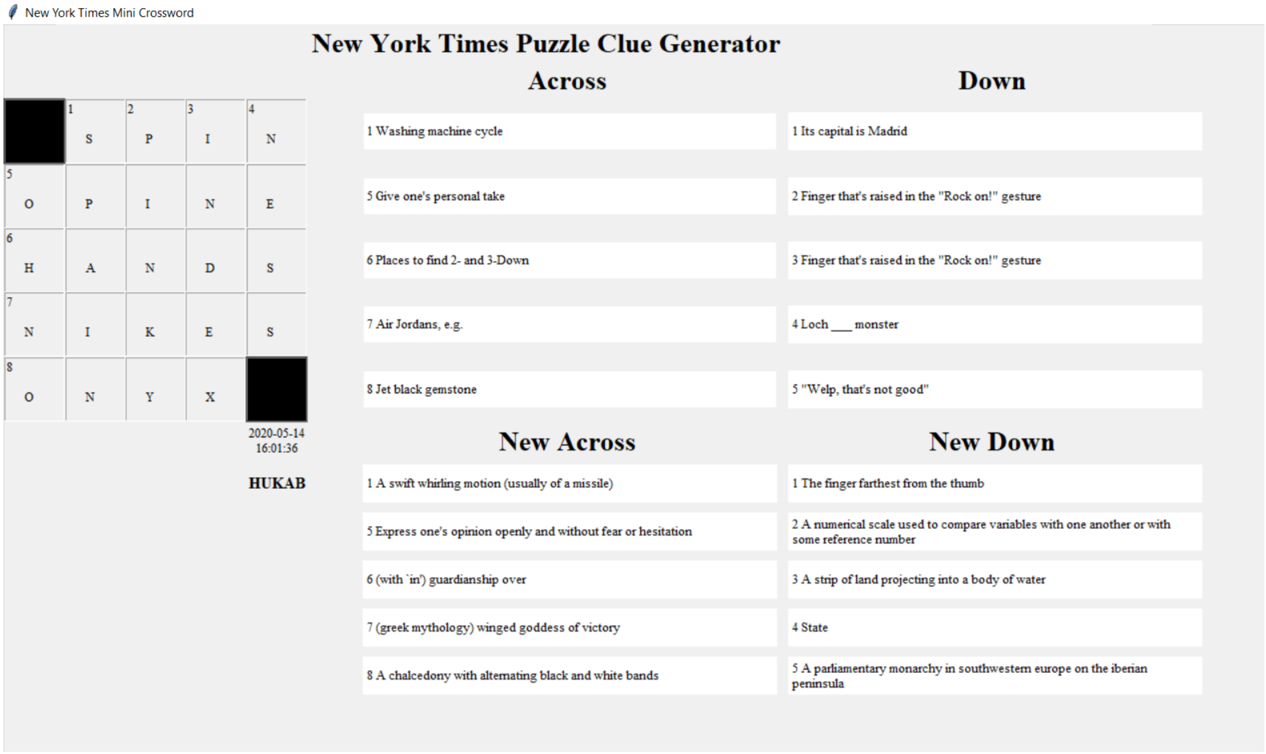
2 Sound of strumming

3 Opinion or judgment

4 An interest followed with exaggerated zeal

5 (astronomy) a relatively small extraterrestrial body consisting of a frozen mass that travels around the sun in a highly elliptical orbit

Test 11



New York Times Mini Crossword

New York Times Puzzle Clue Generator

■	1 S	2 P	3 I	4 N
5 O	P	I	N	E
6 H	A	N	D	S
7 N	I	K	E	S
8 O	N	Y	X	■

2020-05-14 16:01:36

HUKAB

Across

1 Washing machine cycle

5 Give one's personal take

6 Places to find 2- and 3-Down

7 Air Jordans, e.g.

8 Jet black gemstone

Down

1 Its capital is Madrid

2 Finger that's raised in the "Rock on!" gesture

3 Finger that's raised in the "Rock on!" gesture

4 Loch ___ monster

5 "Welp, that's not good"

New Across

1 A swift whirling motion (usually of a missile)

5 Express one's opinion openly and without fear or hesitation

6 (with `in') guardianship over

7 (greek mythology) winged goddess of victory

8 A chalcedony with alternating black and white bands

New Down

1 The finger farthest from the thumb

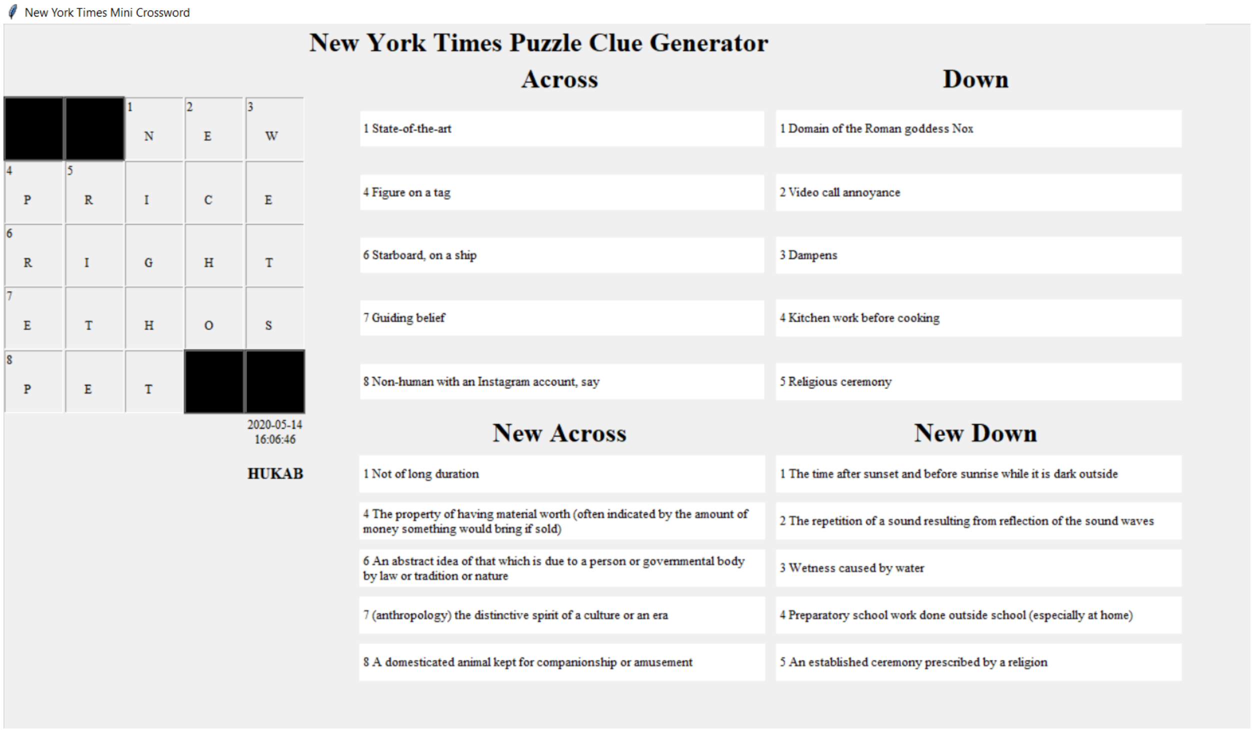
2 A numerical scale used to compare variables with one another or with some reference number

3 A strip of land projecting into a body of water

4 State

5 A parliamentary monarchy in southwestern europe on the iberian peninsula

Test 12



New York Times Mini Crossword

New York Times Puzzle Clue Generator

■	■	1 N	2 E	3 W
4 P	5 R	I	C	E
6 R	I	G	H	T
7 E	T	H	O	S
8 P	E	T	■	■

2020-05-14
16:06:46

HUKAB

Across

1 State-of-the-art

4 Figure on a tag

6 Starboard, on a ship

7 Guiding belief

8 Non-human with an Instagram account, say

Down

1 Domain of the Roman goddess Nox

2 Video call annoyance

3 Dampens

4 Kitchen work before cooking

5 Religious ceremony

New Across

1 Not of long duration

4 The property of having material worth (often indicated by the amount of money something would bring if sold)

6 An abstract idea of that which is due to a person or governmental body by law or tradition or nature

7 (anthropology) the distinctive spirit of a culture or an era

8 A domesticated animal kept for companionship or amusement

New Down

1 The time after sunset and before sunrise while it is dark outside

2 The repetition of a sound resulting from reflection of the sound waves

3 Wetness caused by water

4 Preparatory school work done outside school (especially at home)

5 An established ceremony prescribed by a religion

Test 13

2-Source Code

Part 1-Web Scraping:

```
from selenium import webdriver
from time import sleep
import re
from datetime import *
from tkinter import *
import time
class ExtractPuzzle():
    def __init__(self):
        cpath = r"C:\Users\abdul\Desktop\AIproject\chromedriver.exe"
        self.driver = webdriver.Chrome(cpath)

    def Start(self):
        #EXTRACTING DATA
        self.driver.get('https://www.nytimes.com/crosswords/game/mini')
        sleep(2)
        startBtn = self.driver.find_element_by_xpath('//*[@id="root"]/div/div/div[4]/div/main/div[2]/div/div[2]/div[3]/div/article/div[2]/button/div')
        startBtn.click()
        sleep(2)
        r1 = self.driver.find_element_by_xpath('//*[@id="root"]/div/div/div[4]/div/main/div[2]/div/div/ul/div[2]/li[2]/button')
        r1.click()
        r2 = self.driver.find_element_by_xpath('//*[@id="root"]/div/div/div[4]/div/main/div[2]/div/div/ul/div[2]/li[2]/ul/li[3]/a')
        r2.click()
        sleep(2)
        r3 =
self.driver.find_element_by_xpath('//*[@id="root"]/div/div[2]/div[2]/article/div[2]/button[2]/div')
        r3.click()
        sleep(2)
        c1 = self.driver.find_element_by_xpath('//*[@id="root"]/div/div[2]/div[2]/span')
        c1.click()
        sleep(2)
        element = self.driver.find_element_by_xpath('//*[@id="root"]/div/div/div[4]/div/main/div[2]')
        content = element.get_attribute("innerHTML")
        self.driver.close()
        file = open("file.txt", "w")
        file.write(content)
        file.close()

#main
bot = ExtractPuzzle()
bot.Start()


#Showing solved Puzzle----------------------------------------------
file = open("file.txt", "r")
content = file.read()
file.close()
output = " "
i = 1
numbers = []
while i <= 25:  #the total number of letters
    indexLetter = 'id="cell-id-' + str(i-1) + '"'
    index = content.find(indexLetter)
    letter = content[index + (len(indexLetter) + 8):index + (len(indexLetter) + 18)]

    if letter == 'Cell-block':

        row = int(i / 5)

        col = i % 5

        output += "-      "

        if i % 5 == 0:  #check if 5 have been completed so we move to next line

            output += " \n\n"

        i+=1

        continue

    else:

        index2 = content.find('text-anchor="middle"', index)

        indexStart = content.find('text-anchor="start"', index, index2)

        if indexStart != (-1):

            indexNumber = content.find("</text>", indexStart)

            letterNumber = content[indexNumber + 7]

            output += ""+letterNumber

            numbers.append(letterNumber) # array storing numbers

        else:

             output += " "

        index3 = content.find('</text>',index2)

        mainletter = content[index3-1]

        output += mainletter+"      "


    if i % 5 == 0:   # we need to get to new line after 5

        output += " \n\n"

    i+=1



# print(output)
# till here it's just working on the console

#gui part______________________

l2 = output.split("    ")
#print(l2)
today = date.today()
# print("\n                        "+str(today) )
# print("                        HUKAB")

#displaying the hints____________________________________

u = 1
#print("\n ACROSS:\n")

fifthclue = l2[5]
fifthclue = fifthclue[5]
sixthclue = l2[10]
sixthclue = sixthclue[5]
seventhclue = l2[15]
seventhclue = seventhclue[5]
e8clue = l2[15]
e8clue = e8clue[5]

txt = ["", "1 ", '4 ' , str(sixthclue)+" ", str(seventhclue)+" ", str(seventhclue)+" ", "1 ","2 ","3 ","4 ","5 "]



acrossclues = []
```

```
while u < 11:
    t1 = content.split('Clue-text--3lZl7">') #splitting the html code using the class details to extract text
    a1 = t1[u].split('</span>')

    txt[u] +=  a1[0]
    # print(txt[u])
    acrossclues.append(txt[u])
    u += 1
    if u == 6:
        print("\n DOWN:")

# if(acrossclues[3][0] == acrossclues[4][0]):
#     a = int (acrossclues[4][0]) + 1
#     a = str(a) + acrossclues[4][1:]
#     acrossclues[4] = a
```

Part 2-Clue Generation:

```
import project1 as p1
import numpy as np
import pandas as pd
import re
from textblob import TextBlob as tb
import nltk
from nltk.corpus import wordnet
import selenium.webdriver as swd
import io
def findDefD(word): #find def by webscraping

    cpath = r"C:\Users\abdul\Desktop\AIproject\chromedriver.exe"
    # url2 = 'https://www.dictionary.com/'
    url1 = 'https://www.merriam-webster.com/' #will be using 2 dictionaries to check
    content = ""
    try:

        browser = swd.Chrome(cpath)
        browser.get(url1)
        searchBox = browser.find_element_by_xpath('/html/body/div[1]/div/header/div/div[4]/form/div[1]/input')
        searchBox.send_keys(word)
        searchBox.submit()
        p1.sleep(3)
        element = browser.find_element_by_class_name('vg')
        content = element.get_attribute("innerHTML")
        browser.close()
    except:
        browser.close()

    with io.open("word.txt", "w", encoding="utf-8") as f:
        f.write(content)
        f.close()

    file = open("word.txt", "r")
    content = file.read()
    file.close()
    finaldef = []

    pattern = re.compile(r"<.*?>")
    definations = re.sub(pattern, "", content)
    finaldef = definations.split('\n')

    return finaldef[0]


def findDefW(word):  #find defination from the wordnet
    synset = wordnet.synsets(word)

    # print('Word and Type : ' + synset[0].name())
    # print('Synonym is: ' + synset[0].lemmas()[0].name())

    # print('Example is : ' + str(synset[0].examples()))
    try:
        return( synset[0].definition())
    except:
        return ""




def cleanword(word): #remove spaces and numbers
    p1 = re.compile(r"\s")
    p3 = re.compile(r"\d")
    w = word
    w = re.sub(p1, "", w)
    w = re.sub(p3, "", w)
    w = w.lower()
    return w

def cleanx2(word): #clean again to remove -
    p1 = re.compile(r"\s")
    p2 = re.compile(r"-")
    p3 = re.compile(r"\d")
    w = re.sub(p2, "", word)
    w = re.sub(p1, "", w)
    w = re.sub(p3, "", w)
    w = w.lower()
    return w

def compile_down_words(o): #compiles the down words
    dwords = []
    for i in range(5):
        word = ""
        for j in range(5):
            word  += o[j][i]
        dwords.append(word)
    return dwords



def compile_words(o): # compiles across and down words

    across = o.split('\n')

    awords = [] # Across words according to their index-1
    dwords = [] # Down words according to their index-1
    # Across word extraction
    for i in range(len(across)):
        w = cleanword(across[i])
        if w != '':
            awords.append(w)
    #Down word extraction
    dwords = compile_down_words(awords)
    for i in range(5):
        aw = cleanx2(awords[i])
        dw = cleanx2(dwords[i])
        awords[i] = aw
        dwords[i] = dw
    return awords, dwords


#__________________________________________________________________________ compiling part

def clean_clue(clue):
    p1 = re.compile(r";|\.|:")
    w = re.sub(p1, ";", clue)
    neww = w.split(";")

    return neww[0]

def find_new_clue(word):

    clue = findDefW(word)

    if clue == '':
        clue = findDefD(word)

    clue = clean_clue(clue) # making the clue short

    clue = clue.capitalize()

    return clue



def generate_clues_all(awords, dwords):
    new_across_clues = []
    new_down_clues = []

    for i in range(5):
        new_across_clues.append(find_new_clue(awords[i]))
        new_down_clues.append(find_new_clue(dwords[i]))

    return new_across_clues, new_down_clues

# nltk.download('wordnet')

awords, dwords = compile_words(p1.output)
new_across_clues, new_down_clues = generate_clues_all(awords, dwords)

print(p1.output)
print(p1.acrossclues)

print("Across words", awords)
print("Down words", dwords)
print("New Across words", new_across_clues)
print("New Down words", new_down_clues)
# print(p1.acrossclues)   #all the clues in this array
# print(p1.output)  #the puzzle in txt form
```

Part 3-Graphical User Interface (GUI):

```
import project1 as p1
import project2 as p2


#GUI INTERFACE
root = p1.Tk()
root.geometry('800x800') #resolution
label_1= p1.Label(root,text="New York Times Mini Puzzle", font=('Times New Roman','20')) #heading
label_1.grid(row=0,column = 0, columnspan = 5)


# #Creating border color
# Border = LabelFrame(root,bd=5, bg="blue", relief=FLAT)
# Border.pack(padx=10, pady=10)
u = 0
r = 1 #row
col = 0 #column
while u <25:
#button 1 statisfying both the possibility of black box and clue box
    if col < 5:
        if p1.l2[u][-1] == '-': #add black block
            button_u = p1.Button(root, bg="black", fg ="black",bd =2, width=8, height=4, font=('Times New Roman', '13'))
            button_u.grid(row=r,column=col)
        elif p1.l2[u][-2] != '':
            button_u = p1.Button(root, text = p1.l2[u][-2] + '\n' + '\n'  + '      ' + p1.l2[u][-1], bd = 1 ,fg="black", width=8, height=4, anchor= p1.NW, font=('Times New Roman', '13'), justify = p1.LEFT)
            button_u.grid(row=r,column=col)
        else:
            button_u = p1.Button(root, text = p1.l2[u][-1],  fg="black", width=8, height=4, font=('Times New Roman', '13'))
            button_u.grid(row=r,column=col)
        col += 1
        u += 1
    else:
        col = 0
        r += 1


now = p1.datetime.now()
current_time = now.strftime("%H:%M:%S")
today = str(p1.today) +"\n"+ str(current_time)
button_cluesacross = p1.Button(root, text = "ORIGINAL CLUES:" + '\n' +  p1.acrossclues[0] +'\n' + p1.acrossclues[1] + '\n' + p1.acrossclues[2] +'\n' + p1.acrossclues[3] + '\n' + p1.acrossclues[4] + '\n' + '\n' + "NEW ACROSS:"+ '\n' + "1 " + p2.new_across_clues[0] +'\n' + "4 " + p2.new_across_clues[1] + '\n' + "6 " + p2.new_across_clues[2] + '\n' + "7 " + p2.new_across_clues[3] + '\n' + "8 " + p2.new_across_clues[4] , fg="Black", width=21,bg ="white", height=25, font=('Times New Roman', '12'),wraplength = 190, anchor = p1.NW, justify = p1.LEFT, padx=2)
button_cluesacross.grid(row=1,column=5, rowspan = 6)
myLabel1= p1.Label(root, text = 'Across', font=('Times New Roman', '20'))
myLabel1.grid(row=0,column=5)
button_cluesdown = p1.Button(root, text = "ORIGINAL CLUES:" + '\n' + p1.acrossclues[5] +'\n' + p1.acrossclues[6] + '\n' + p1.acrossclues[7] +'\n' + p1.acrossclues[8] + '\n' + p1.acrossclues[9] + '\n' + '\n' + "NEW DOWN:" + '\n' + "1 " + p2.new_down_clues[2]+ '\n' + "2 " + p2.new_down_clues[3]+ '\n' + "3 " + p2.new_down_clues[4]+ '\n' + "4 " + p2.new_down_clues[0]+ '\n' + "5 " + p2.new_down_clues[1], fg="Black", width=21, bg= "white", height=25, font=('Times New Roman', '12'), wraplength = 180, anchor = p1.NW, justify = p1.LEFT, padx=2)
button_cluesdown.grid(row=1,column=6, rowspan = 6)
```

```
myLabel2= p1.Label(root, text = "Down", font=('Times New Roman', '20'))
myLabel2.grid(row=0,column=6)

myLabel3 = p1.Label(root, text = today,  font=('Times New Roman', '12'))
myLabel3.grid(row=6,column=4)
myLabel4 = p1.Label(root, text = "HUKAB", font=('Times New Roman', '12'))
myLabel4.grid(row=7,column=4)
root.mainloop()
```

References:

[1] New York Times, "The Mini Crossword," 14 May 2020. [Online]. Available: https://www.nytimes.com/crosswords/game/mini.

[2] "Selenium with Python," Selenium, [Online]. Available: https://selenium-python.readthedocs.io/ [Accessed 14 May 2020].

[3] " Python GUI Programming With Tkinter," Python, [Online]. Available: https://realpython.com/python-gui-tkinter/ [Accessed 7 May 2020].

[4] Princeton University, "WordNet: A Lexical Database for English." [Online]. Available: https://wordnet.princeton.edu/. [Accessed 14 May 2020]

[5] " Natural Language Toolkit," NLTK, [Online]. Available: https://www.nltk.org/. [Accessed 14 May 2020].

[6] Merriam Webster, [Online]. Available: https://www.merriam-webster.com/. [Accessed 14 May 2020].