**Typing speed and accuracy test**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**G.Karthikeya Reddy 1602-20-737-138**

**N.Bhuvan Swaroop 1602-20-737-128**

**K.Saketh 1602-20-737-165**

****

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)**

**(AFFILIATED TO OSMANIA UNIVERSITY)**

**HYDERABAD - 500 030**

**Department of Information Technology**

****

**DECLARATION BY CANDIDATE**

We, **G.Karthikeya Reddy , N.Bhuvan Swaroop , K.Saketh ,** bearing hall ticket number**,1602-20-737-138,1602-20-737-128,1602-20-737-165** hereby declare that the project report entitled **”Typing speed and accuracy test”** Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project report has not been submitted to any other university or institute for the award of any other degree or diploma.

**G.Karthikeya Reddy 1602-20-737-138**

**N.Bhuvan Swaroop 1602-20-737-128**

**K.Saketh 1602-20-737-165**

**VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)**

**(AFFILIATED TO OSMANIA UNIVERSITY)**

**HYDERABAD - 500 030**

**Department of Information Technology**

****

**BONAFIDE CERTIFICATE**

This is to certify that the project entitled “**Typing speed and accuracy test**” being submitted by **G.Karthikeya Reddy, N.Bhuvan Swaroop, K.Saketh** bearing **1602-20-737-138, 1602-20-737-128, 1602-20-737-165**, in partial fulfillment of the requirements for the completion of MINI PROJECT of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

Dr. K. Ram Mohan Rao

Faculty I/C HOD, IT

**ACKNOWLEDGEMENT**

We thank the department of INFORMATION TECHNOLOGY, for introducing the subject “Mini Project-1” in BE Third Semester.

We would also like to show our appreciation to our Honorable principal, Dr S V Ramana sir, our HOD K. Ram Mohan Rao for supporting us and our mini project guide, **Sireesha Chitteu Mam** for letting us properly understand the process of doing a project and for providing valuable insight and expertise that has greatly assisted us in the making of the project.

**ABSTRACT**

This project aims to design and create media application that will be used to train the speed and accuracy of typing. As we are of IT branch most of the time, we practice a lot of coding on laptop or pc, even in the future after our company placement we almost work about 40+ hours in a company.

Since most of the programming languages are case sensitive even missing the capitals, semi colon, etc. gives an error. This becomes a problem if a code is huge finding the error becomes difficult. So having accuracy and staying focused while typing is important. Even if there was at least 5% increase in typing speed, for a 40+ hours of work the amount of time saved is a lot.

Main features this project includes are using a GUI displaying your accuracy, speed, storing data and giving your stats.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **CONTENTS** | **Page No.** |
| **1** | **Introduction** | **7** |
| **2** | **Technology** | **9** |
| **3** | **Design** | **10** |
| **4** | **Implementation** | **14** |
| **5** | **Results** | **49** |
| **6** | **Additional Knowledge acquired** | **57** |
| **7** | **Conclusion and Future work** | **58** |
| **8** | **References** | **59** |

**INTRODUCTION**

**PURPOSE**

This application mainly aims to decrease the user mistakes while typing, it can either be a code or a story a person is trying to type. Even the smallest mistake while coding gives a lot of errors. By consistently using this application you can understand how much you have improved in your typing.

**INTENDED AUDIENCE**

The intended audience are students pursuing branches that involves coding, data entry clerk and typists.

**PRODUCT SCOPE**

In our application a sentence or sentences (depending on your choice) will be displayed and the user has to enter the given sentence. According to your errors and speed the output will be displayed. You can join a test that has been created previously or start a practice test where sentences are generated randomly.

**PROBLEM DEFINITION**

Lack of instructional media typing speed appropriate to the circumstances in the classroom was the background of this project. The problem of this study was to determine the improvement rate at which typing speed and accuracy were achieved using computer-based training.

**TECHNOLOGY**

**Hardware Requirements**

* 512 MB RAM
* 2GB HDD
* CORE i5

**Software Requirements**

* Windows XP/Windows 2000
* PYTHON Interpreter

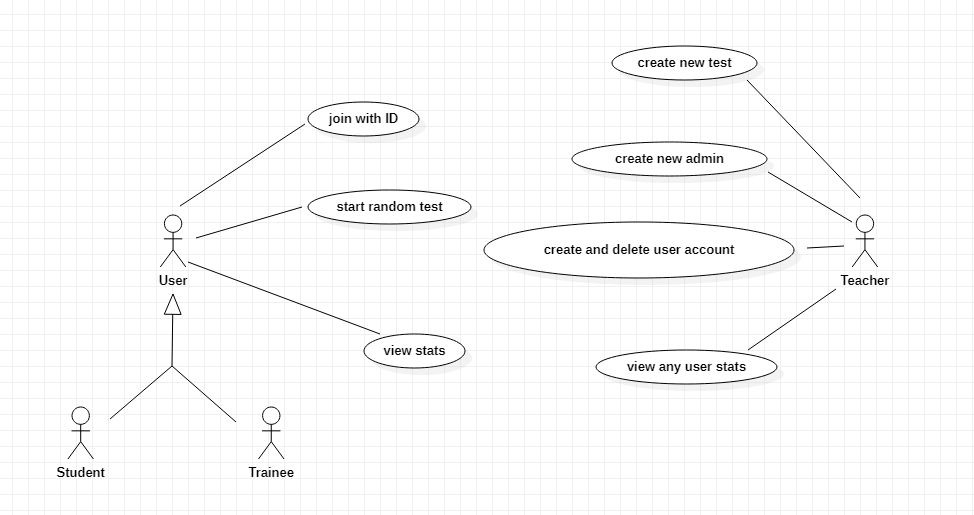
**Packages used**

* pygame
* sys
* os
* time
* random
* essential\_generators

**PROPOSED WORK**

**DESIGN**

Use case diagram



**USE CASES**

Actors: User

Name: start random test

Description: allows the user to start a random test that is not created by admin

and not stored in his stats.

Main Flow:

|  |  |
| --- | --- |
| **User**  **1.Chooses random test option.**  **3.Enters the input accordingly.** | **System**  **2.Displays sentences randomly.**  **4.Displays the speed and accuracy.** |

Actors: Teacher

Name: create new test

Description: allows the teacher to create a new test, add sentences and difficulty to it.

Main Flow:

|  |  |
| --- | --- |
| **Teacher**  **1.Selects create new test.**  **3.Enters the correct details.**  **5.Gives few sentences as input and gives id for the exam.** | **System**  **2.Asks the teacher to login with admin id and password.**  **4.Logged in successfully, asks to enter the sentences to add for the test.**  **6.The data is stored with the given id.** |

Actors: User

Name: view stats

Description: displays stats showing your progression after logging with your ID.

Main Flow:

|  |  |
| --- | --- |
| **User**  **1.Selects view stats.**  **3.Enter the details correctly.** | **System**  **2.Prompts the user to enter his login id and password.**  **4.Displays the stats of the user.** |

**ACTIVITY DIAGRAM**

Take input for execution

Set all panels and buttons

Get exercise

Check time

Enter key

False

Increment invalid counter

valid character

True

Increment in correct counter

Show results

**IMPLEMENTATION**

**Modules**

pygame : For design and user interface.

sys : to manipulate different parts of python runtime environment.

os : provides functions for interacting with the operating system.

time : provides functions for calculating time at certain part of code.

random : an in-built module of Python which is used to generate random numbers.

essential\_generators : This is the built-in module in python, and this module is useful to generate a random sentence, word, paragraph, etc.

**Source code**

**#**main code

import pygame

from pygame.locals import \*

import sys

import os

import time

import random

from essential\_generators import DocumentGenerator

class Game:

def \_\_init\_\_(self):

self.w=1920

self.h=1080

self.reset=True

self.active = False

self.input\_text=''

self.word = ''

self.time\_start = 0

self.total\_time = 0

self.accuracy = 0

self.results = 'Time:0 Accuracy:0 % Wpm:0 '

self.results\_1= 'adjusted wpm:0 errors:0'

self.wpm = 0

self.awpm = 0

self.e = 0

self.e1 = 0

self.n=5

self.end = False

self.HEAD\_C = (255,213,102)

self.TEXT\_C = (240,240,240)

self.RESULT\_C = (255,70,70)

self.t = ''

self.k = ''

self.id=''

self.passw=''

self.rpassw=''

self.stud=''

self.fi=0

pygame.init()

self.open\_img = pygame.image.load('type-speed-open.jpg')

self.open\_img = pygame.transform.scale(self.open\_img, (self.w,self.h))

self.bg2 = pygame.image.load('background3.jpg')

self.bg2 = pygame.transform.scale(self.bg2, (1920,1080))

self.bg = pygame.image.load('background.jpg')

self.bg = pygame.transform.scale(self.bg, (1920,1080))

self.back = pygame.image.load('back.png')

self.back = pygame.transform.scale(self.back, (125,75))

self.screen = pygame.display.set\_mode((self.w,self.h))

pygame.display.set\_caption('Type Speed test')

def draw\_text(self, screen, msg, y ,fsize, color):

font = pygame.font.Font(None, fsize)

text = font.render(msg, 1,color)

text\_rect = text.get\_rect(center=(self.w/2, y))

screen.blit(text, text\_rect)

pygame.display.update()

def draw\_text1(self, screen, msg, x, y ,fsize, color):

font = pygame.font.Font(None, fsize)

text = font.render(msg, 1,color)

text\_rect = text.get\_rect(center=(x, y))

screen.blit(text, text\_rect)

pygame.display.update()

def get\_sentence(self):

f = open('sentences.txt').read()

sentences = f.split('\n')

sentence = random.choice(sentences)

return sentence

def calc(self, screen):

if(not self.end):

#Calculate time

self.total\_time += time.time() - self.time\_start

#Calculate accuracy

'''count = 0

for i,c in enumerate(self.word):

try:

if self.input\_text[i] == c:

count += 1

except:

pass'''

#self.accuracy = count/len(self.word)\*100

self.error()

#Calculate words per minute

self.wpm += len(self.input\_text)\*60/(5\*self.total\_time)

self.accuracy += (len(self.word)-self.e)/len(self.word)\*100

if(len(self.input\_text)==0):

self.accuracy+=0

self.end = True

pygame.display.update()

def show\_results(self):

self.total\_time/=self.n

self.wpm/=self.n

self.accuracy/=self.n

self.awpm=self.wpm-((self.e\*60)/(self.total\_time))

self.results = 'Time:'+str(round(self.total\_time)) +" secs Accuracy:"+ str(round(self.accuracy)) + "%" + ' Wpm: ' + str(round(self.wpm))

self.results\_1= 'adjusted wpm: ' + str(round(self.awpm)) + ' errors:' + str(self.e1)

def run(self):

self.running=True

self.reset=False

self.end=False

self.input\_text=''

msg = "Typing Speed Test"

self.draw\_text(self.screen, msg,80, 80,self.HEAD\_C)

self.screen.blit(self.back,(10,60))

pygame.display.update()

while(self.running):

limit=0

clock = pygame.time.Clock()

self.screen.fill((0,0,0), (550,250,820,50))

pygame.draw.rect(self.screen,self.HEAD\_C, (550,250,820,50), 2)

# update the text of user input

self.draw\_text(self.screen, self.input\_text, 274, 26,(250,250,250))

if(len(self.input\_text)>70):

limit=1

pygame.display.update()

pygame.display.update()

for event in pygame.event.get():

if event.type == QUIT:

self.running = False

pygame.display.quit()

pygame.quit()

quit()

#sys.exit()

elif event.type == pygame.MOUSEBUTTONUP:

x,y = pygame.mouse.get\_pos()

# position of input box

if(x>=550 and x<=1370 and y>=250 and y<=300):

self.active = True

self.input\_text = ''

self.time\_start = time.time()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

elif event.type == pygame.KEYDOWN:

if self.active and not self.end:

if event.key == pygame.K\_RETURN:

self.calc(self.screen)

self.end = True

self.running=False

break

elif event.key == pygame.K\_BACKSPACE:

self.input\_text = self.input\_text[:-1]

limit=0

else:

if(limit!=1):

self.input\_text += event.unicode

pygame.display.update()

clock.tick(60)

def reset\_game(self):

self.screen.blit(self.open\_img, (0,0))

pygame.display.update()

time.sleep(3)

self.reset=False

self.end = False

self.active=False

self.input\_text=''

self.word = ''

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

msg = "Typing Speed Test"

self.draw\_text(self.screen, msg,80, 80,self.HEAD\_C)

pygame.draw.rect(self.screen,(255,192,25), (650,250,650,50), 2)

self.screen.blit(self.back,(10,60))

pygame.display.update()

def start(self):

self.screen.fill((0,0,0))

self.screen.blit(self.bg2,(0,0))

self.draw\_text(self.screen, "Create new test",125, 35,self.TEXT\_C)

self.draw\_text(self.screen, "Start a random test",225, 35,self.TEXT\_C)

self.draw\_text(self.screen, "Join with test ID",325, 35,self.TEXT\_C)

self.draw\_text(self.screen, "View stats",425, 35,self.TEXT\_C)

self.draw\_text(self.screen, "Create new student account",525, 35,self.TEXT\_C)

self.draw\_text(self.screen, "Create new administrator",625, 35,self.TEXT\_C)

self.draw\_text(self.screen, "Delete student account",725, 35,self.TEXT\_C)

self.draw\_text(self.screen, "Exit",825, 35,self.TEXT\_C)

pygame.display.update()

self.running=True

while(self.running):

for event in pygame.event.get():

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=870 and x<=1060 and y>=100 and y<=150):

self.create\_test()

elif(x>=850 and x<=1080 and y>=200 and y<=250):

self.random()

elif(x>=870 and x<=1060 and y>=300 and y<=350):

self.join\_id()

elif(x>=890 and x<=1040 and y>=400 and y<=450):

self.stats()

elif(x>=810 and x<=1120 and y>=500 and y<=550):

self.create\_student()

elif(x>=820 and x<=1110 and y>=600 and y<=650):

self.create\_admin()

elif(x>=820 and x<=1110 and y>=700 and y<=750):

self.delete\_student()

elif(x>=900 and x<=990 and y>=800 and y<=850):

self.running = False

pygame.display.quit()

pygame.quit()

sys.exit()

def delete\_student(self):

self.fi=0

self.authorization()

self.id=''

self.passw=''

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.blit(self.back,(10,60))

self.draw\_text1(self.screen, 'Enter student id to be deleted:',700, 250, 26,(250,250,250))

self.displaying()

try:

with open('student.txt','r') as f1:

lines = f1.readlines()

with open('student.txt','w') as f1:

for line in lines:

first=0

for word in line.split():

if(word==self.id and first==0):

first=-1

break

first+=1

if(first!=-1):

f1.write(line)

first+=1

with open('student.txt','r') as f1:

lines = f1.readlines()

with open('student.txt', 'w') as f1:

for line in lines:

if line.strip("\n") != self.id:

f1.write(line)

os.remove(self.id+'.txt')

self.draw\_text(self.screen,'Successfully removed',600, 28, self.RESULT\_C)

pygame.display.update()

time.sleep(2)

self.start()

except FileNotFoundError:

self.draw\_text(self.screen,'User does not exist',600, 28, self.RESULT\_C)

pygame.display.update()

time.sleep(2)

self.start()

def create\_test(self):

self.fi=0

self.authorization()

self.id=''

self.passw=''

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.blit(self.back,(10,60))

self.draw\_text1(self.screen, 'Enter test id:',800, 250, 26,(250,250,250))

pygame.display.update()

self.displaying()

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.blit(self.back,(10,60))

f1=open(self.id+'.txt','a')

self.draw\_text1(self.screen, 'Enter length of the test:',800, 250, 26,(250,250,250))

self.displaying()

f1.write(self.id+'\n')

try:

self.n=int(self.id)

except ValueError:

self.draw\_text(self.screen,'Invalid! please enter a number',600, 28, self.RESULT\_C)

time.sleep(2)

pygame.display.update()

self.create\_test()

for i in range(self.n):

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.blit(self.back,(10,60))

temp=str(i+1)

if(i==0):

self.draw\_text(self.screen,'Enter '+temp+'st sentence:',200, 28, (250,250,250))

elif(i==1):

self.draw\_text(self.screen,'Enter '+temp+'nd sentence:',200, 28, (250,250,250))

elif(i==2):

self.draw\_text(self.screen,'Enter '+temp+'rd sentence:',200, 28, (250,250,250))

else:

self.draw\_text(self.screen,'Enter '+temp+'th sentence:',200, 28, (250,250,250))

self.displaying1()

f1.write(self.id)

f1.write('\n')

self.draw\_text(self.screen,'Test has been created successfully',800, 28, self.RESULT\_C)

time.sleep(2)

f1.close()

self.start()

def displaying1(self):

self.id=''

self.running=True

self.end=True

while(self.running):

limit=0

self.screen.fill((0,0,0), (550,250,820,50))

pygame.draw.rect(self.screen,(255,192,25), (550,250,820,50), 2)

self.draw\_text(self.screen, self.id,274, 26,(250,250,250))

if(len(self.id)>70):

limit=1

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RETURN:

self.running=False

self.end=False

elif event.key == pygame.K\_BACKSPACE:

limit=0

self.id = self.id[:-1]

else:

if(limit!=1):

self.id += event.unicode

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

if self.end:

break

def displaying(self):

self.id=''

self.running=True

self.end=True

while(self.running):

limit=0

self.screen.fill((0,0,0), (910,225,200,50))

pygame.draw.rect(self.screen,self.HEAD\_C, (910,225,200,50), 2)

self.draw\_text1(self.screen, self.id,1010, 250, 26,(250,250,250))

if(len(self.id)>16):

limit=1

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RETURN:

self.running=False

self.end=False

elif event.key == pygame.K\_BACKSPACE:

self.id = self.id[:-1]

limit=0

else:

if(limit!=1):

self.id += event.unicode

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

if self.end:

break

def random(self):

self.e1=0

self.time\_start = 0

self.total\_time = 0

self.accuracy = 0

self.wpm = 0

self.awpm = 0

self.e=0

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.blit(self.back,(10,60))

self.screen.fill((0,0,0), (910,250,100,50))

self.n=5

pygame.draw.rect(self.screen,(255,192,25), (910,250,100,50), 2)

self.draw\_text(self.screen, "Enter length of the test",200, 28,self.TEXT\_C)

self.running=True

self.end=True

while(self.running):

self.screen.fill((0,0,0), (910,250,100,50))

pygame.draw.rect(self.screen,self.HEAD\_C, (910,250,100,50), 2)

self.draw\_text(self.screen, self.t, 274, 26,(250,250,250))

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RETURN:

self.running=False

self.end=False

elif event.key == pygame.K\_BACKSPACE:

self.t = self.t[:-1]

else:

self.t += event.unicode

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

if self.end:

break

pygame.display.update()

try:

self.n=int(self.t)

except ValueError:

self.draw\_text(self.screen,'Invalid! please enter a number',600, 28, self.RESULT\_C)

time.sleep(2)

pygame.display.update()

self.random()

for i in range(self.n):

self.reset\_game()

gen = DocumentGenerator()

self.word = gen.sentence()

if(len(self.word)>70):

self.word=self.word[0:70]

self.draw\_text(self.screen, self.word,200, 28,self.TEXT\_C)

self.run()

self.show\_results()

self.draw\_text(self.screen, self.results,450, 28, self.RESULT\_C)

self.draw\_text(self.screen, self.results\_1,500, 28, self.RESULT\_C)

self.running=True

while(self.running):

for event in pygame.event.get():

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

def join\_id(self):

self.fi=1

self.authorization()

f2=open(self.id+'.txt','a')

self.id=''

self.passw=''

self.screen.blit(self.back,(10,60))

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.blit(self.back,(10,60))

self.draw\_text1(self.screen, 'Enter test id:',800, 250, 26,(250,250,250))

pygame.display.update()

self.displaying()

self.e1=0

self.time\_start = 0

self.total\_time = 0

self.accuracy = 0

self.wpm = 0

self.awpm = 0

self.e=0

cc=0

try:

f1=open(self.id+'.txt')

for line in f1:

if(cc==0):

cc+=1

self.n=int(line)

else:

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.blit(self.back,(10,60))

line = line[:-1]

self.word=line

self.draw\_text(self.screen,line,200, 28, (250,250,250))

self.reset=False

self.end = False

self.active=False

self.run()

self.show\_results()

self.draw\_text(self.screen, self.results,450, 28, self.RESULT\_C)

self.draw\_text(self.screen, self.results\_1,500, 28, self.RESULT\_C)

f2.write(self.results+'\n')

f2.write(self.results\_1+'\n')

pygame.display.update()

f1.close()

f2.close()

self.running=True

while(self.running):

for event in pygame.event.get():

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

except FileNotFoundError:

self.draw\_text(self.screen,'Invalid test id',600, 28, self.RESULT\_C)

pygame.display.update()

time.sleep(2)

self.start()

self.start()

def stats(self):

self.fi=1

self.authorization()

f1=open(self.id+'.txt','r')

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.blit(self.back,(10,60))

c=150

i=0

d=1

p=1

f1.seek(0)

for line in f1:

line = line[:-1]

if(i%2==0):

c+=50

temp=str(d)

if(p>5):

self.draw\_text(self.screen,'Press enter to continue',1000, 28, self.RESULT\_C)

self.running=True

while(self.running):

for event in pygame.event.get():

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RETURN:

self.running=False

break

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.blit(self.back,(10,60))

c=200

p=1

if(d==1):

self.draw\_text(self.screen,temp+'st test',c-50, 28, self.RESULT\_C)

elif(d==2):

self.draw\_text(self.screen,temp+'nd test',c-50, 28, self.RESULT\_C)

elif(d==3):

self.draw\_text(self.screen,temp+'rd test',c-50, 28, self.RESULT\_C)

else:

self.draw\_text(self.screen,temp+'th test',c-50, 28, self.RESULT\_C)

d+=1

p+=1

self.draw\_text(self.screen,line,c, 28, (250,250,250))

c+=50

i+=1

pygame.display.update()

pygame.display.update()

self.running=True

while(self.running):

for event in pygame.event.get():

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

f1.close()

def login(self):

self.screen.blit(self.back,(10,60))

self.running=True

self.end=True

while(self.running):

limit=0

self.screen.fill((0,0,0), (910,225,200,50))

self.screen.fill((0,0,0), (910,325,200,50))

pygame.draw.rect(self.screen,self.HEAD\_C, (910,225,200,50), 2)

pygame.draw.rect(self.screen,self.HEAD\_C, (910,325,200,50), 2)

self.draw\_text1(self.screen, self.id,1010, 250, 26,(250,250,250))

pygame.display.update()

if(len(self.id)>16):

limit=1

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RETURN:

self.running=False

self.end=False

elif event.key == pygame.K\_BACKSPACE:

self.id = self.id[:-1]

limit=1

else:

if(limit!=1):

self.id += event.unicode

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

if self.end:

break

self.running=True

self.end=True

self.k=''

while(self.running):

limit=0

self.screen.fill((0,0,0), (910,325,200,50))

pygame.draw.rect(self.screen,self.HEAD\_C, (910,325,200,50), 2)

self.draw\_text1(self.screen,self.k,1010,350, 26,(250,250,250))

if(len(self.k)>20):

limit=1

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RETURN:

self.running=False

self.end=False

elif event.key == pygame.K\_BACKSPACE:

self.passw = self.passw[:-1]

self.k = self.k[:-1]

limit=0

else:

if(limit!=1):

self.passw += event.unicode

self.k += '\*'

if self.end:

break

pygame.display.update()

def authorization(self):

self.screen.blit(self.back,(10,60))

self.running=True

self.end=True

self.id=''

self.passw=''

self.k=''

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

if(self.fi==0):

self.draw\_text(self.screen,'Admin authorization required',80, 80,self.HEAD\_C)

else:

self.draw\_text(self.screen,'Student login',80, 80,self.HEAD\_C)

self.draw\_text1(self.screen, "Enter id:",800,250, 28,self.TEXT\_C)

self.draw\_text1(self.screen, "Enter password:",800,350, 28,self.TEXT\_C)

pygame.draw.rect(self.screen,self.HEAD\_C, (910,225,200,50), 2)

pygame.draw.rect(self.screen,self.HEAD\_C, (910,325,200,50), 2)

self.login()

f=0

if(self.fi==1):

tf='student.txt'

else:

tf='admin.txt'

with open(tf,'r') as file:

for line in file:

f=0

for word in line.split():

if(f==1):

if(word==self.passw):

self.draw\_text(self.screen,'Authorization successfull!',600, 28, self.RESULT\_C)

pygame.display.update()

time.sleep(2)

f=2

break

if(word==self.id):

self.stud=word

f=1

else:

break

if(f==2):

break

if(f!=2):

self.draw\_text(self.screen,'Invalid Details',600, 28, self.RESULT\_C)

pygame.display.update()

time.sleep(2)

self.start()

pygame.display.update()

def create\_student(self):

self.fi=0

self.authorization()

self.id=''

self.passw=''

self.rpassw=''

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.fill((0,0,0), (910,250,100,50))

f1=open('student.txt','a')

self.draw\_text(self.screen,'Student Details',80, 80,self.HEAD\_C)

self.draw\_text1(self.screen, "Enter id:",800,250, 28,self.TEXT\_C)

self.draw\_text1(self.screen, "Enter password:",800,350, 28,self.TEXT\_C)

self.draw\_text1(self.screen, "Re-Enter password:",800,450, 28,self.TEXT\_C)

pygame.draw.rect(self.screen,self.HEAD\_C, (910,425,200,50), 2)

self.login()

self.running=True

self.end=True

self.k = ''

while(self.running):

limit=0

self.screen.fill((0,0,0), (910,425,200,50))

pygame.draw.rect(self.screen,self.HEAD\_C, (910,425,200,50), 2)

self.draw\_text1(self.screen, self.k, 1010,450, 26,(250,250,250))

if(len(self.k)>20):

limit=1

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RETURN:

self.running=False

self.end=False

elif event.key == pygame.K\_BACKSPACE:

self.rpassw = self.rpassw[:-1]

self.k = self.k[:-1]

limit=0

else:

if(limit!=1):

self.rpassw += event.unicode

self.k += '\*'

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

if self.end:

break

if(self.rpassw != self.passw):

self.draw\_text(self.screen,'Given password and re-entered password must match!',600, 28, self.RESULT\_C)

time.sleep(2)

self.create\_student()

self.draw\_text(self.screen,'data added successfully',600, 28, self.RESULT\_C)

f1.write(self.id+' '+self.passw)

f1.write('\n')

f1.close()

f1=open(self.id+'.txt','a')

f1.close()

time.sleep(2)

self.start()

def create\_admin(self):

self.fi=0

self.authorization()

self.id=''

self.passw=''

self.rpassw=''

self.screen.fill((0,0,0))

self.screen.blit(self.bg,(0,0))

self.screen.fill((0,0,0), (910,250,100,50))

f1=open('admin.txt','a')

f1.write('\n')

self.draw\_text(self.screen,'Administrator Details',80, 80,self.HEAD\_C)

self.draw\_text1(self.screen, "Enter id:",800,250, 28,self.TEXT\_C)

self.draw\_text1(self.screen, "Enter password:",800,350, 28,self.TEXT\_C)

self.draw\_text1(self.screen, "Re-Enter password:",800,450, 28,self.TEXT\_C)

pygame.draw.rect(self.screen,self.HEAD\_C, (910,425,200,50), 2)

self.login()

self.running=True

self.end=True

self.k = ''

while(self.running):

limit=0

self.screen.fill((0,0,0), (910,425,200,50))

pygame.draw.rect(self.screen,self.HEAD\_C, (910,425,200,50), 2)

self.draw\_text1(self.screen, self.k, 1010,450, 26,(250,250,250))

if(len(self.k)>20):

limit=1

pygame.display.update()

for event in pygame.event.get():

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_RETURN:

self.running=False

self.end=False

elif event.key == pygame.K\_BACKSPACE:

self.rpassw = self.rpassw[:-1]

self.k = self.k[:-1]

limit=0

else:

if(limit!=1):

self.rpassw += event.unicode

self.k += '\*'

if event.type == pygame.MOUSEBUTTONDOWN:

x,y = pygame.mouse.get\_pos()

if(x>=10 and x<=135 and y>=60 and y<=110):

self.start()

if self.end:

break

if(self.rpassw != self.passw):

self.draw\_text(self.screen,'Given password and re-entered password must match!',600, 28, self.RESULT\_C)

time.sleep(2)

self.create\_student()

self.draw\_text(self.screen,'data added successfully',600, 28, self.RESULT\_C)

f1.write(self.id+' '+self.passw)

f1.close()

time.sleep(2)

self.start()

f1.close()

def error(self):

j,f,a=660,0,0

self.e=0

for i in range(len(self.input\_text)):

j+=10

if(i>=len(self.word)):

f=1

a=i

break

if(self.input\_text[i]!=self.word[i]):

self.draw\_text1(self.screen, self.input\_text[i], j, 350, 25, self.RESULT\_C)

self.e+=1

else:

self.draw\_text1(self.screen, self.input\_text[i], j, 350, 25, self.TEXT\_C)

if(i<len(self.word)-1):

self.e+=len(self.word)-1-i

if(f==1):

for i in range(a,len(self.input\_text)):

j+=10

self.e+=1

self.draw\_text1(self.screen, self.input\_text[i], j, 350, 25, self.RESULT\_C)

self.e1+=self.e

pygame.display.update()

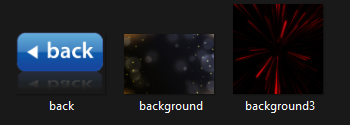
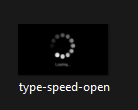
time.sleep(2)

Game().start()

**Github link**

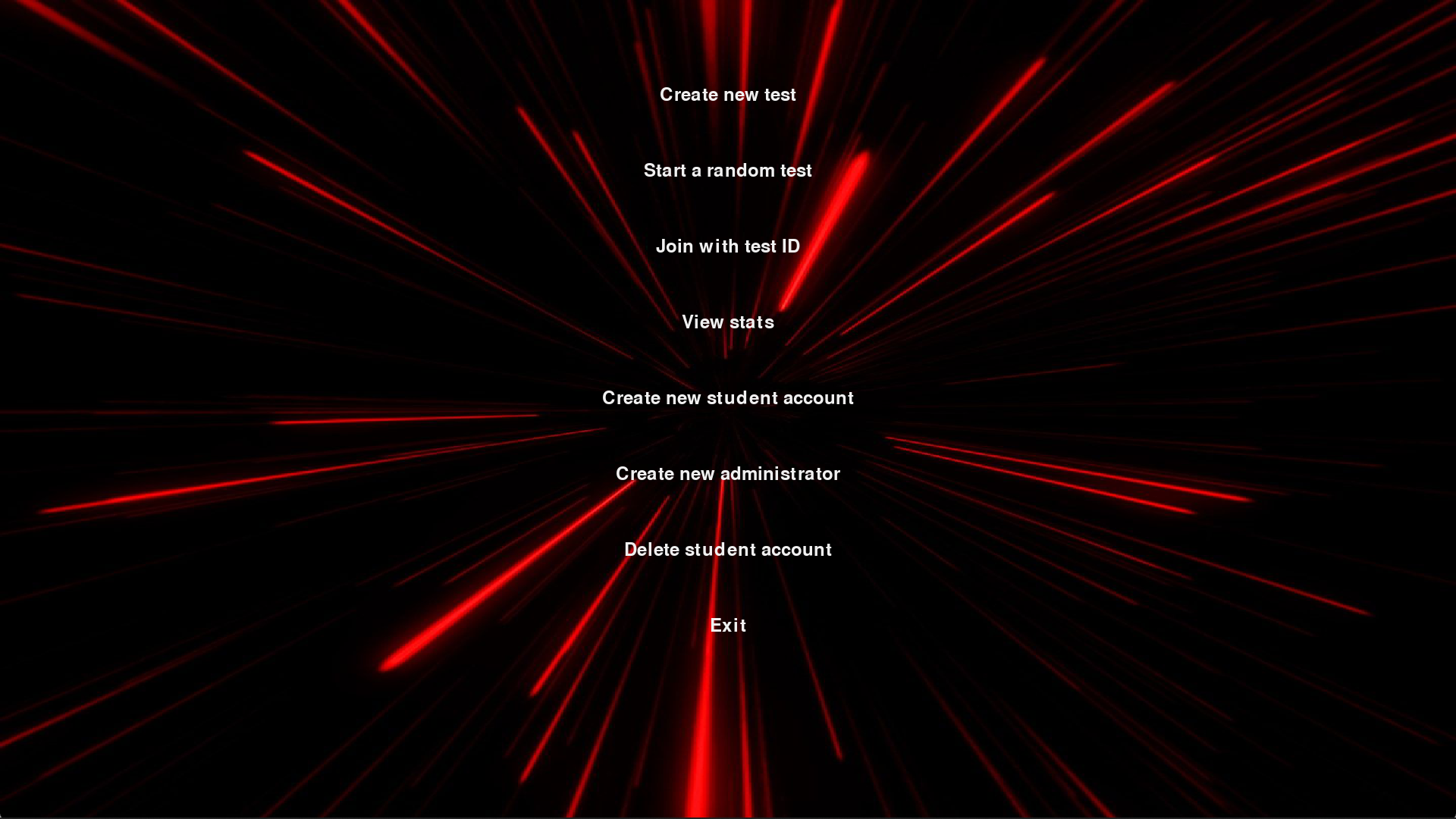
<https://github.com/karthikeya-reddy011/Typing-speed-and-accuracy-test>

**Additional files**

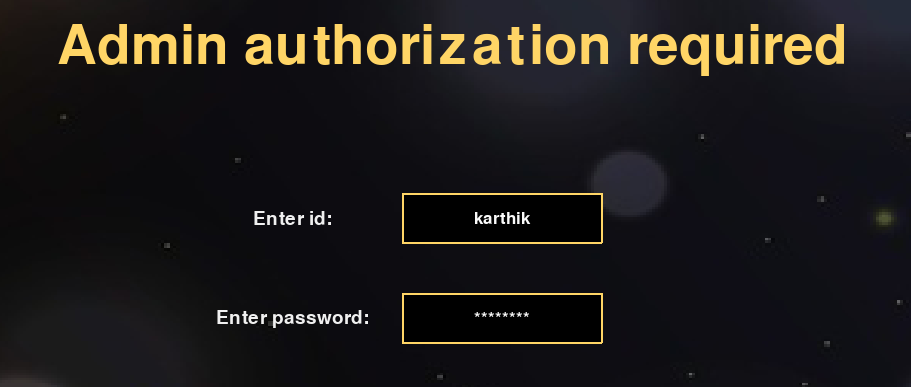
 

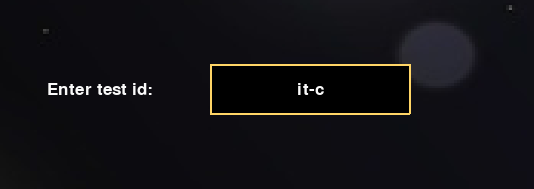
**RESULTS**

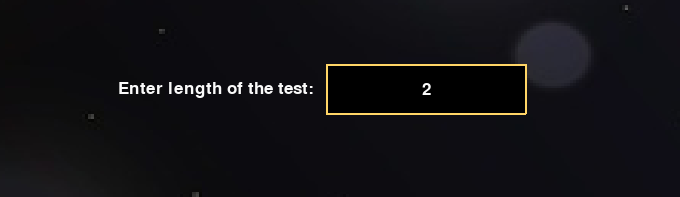
Main menu

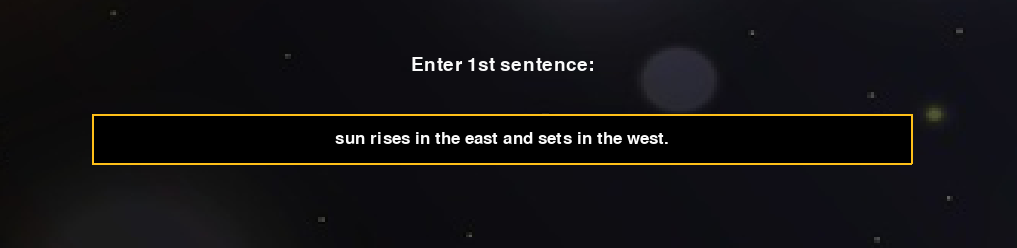


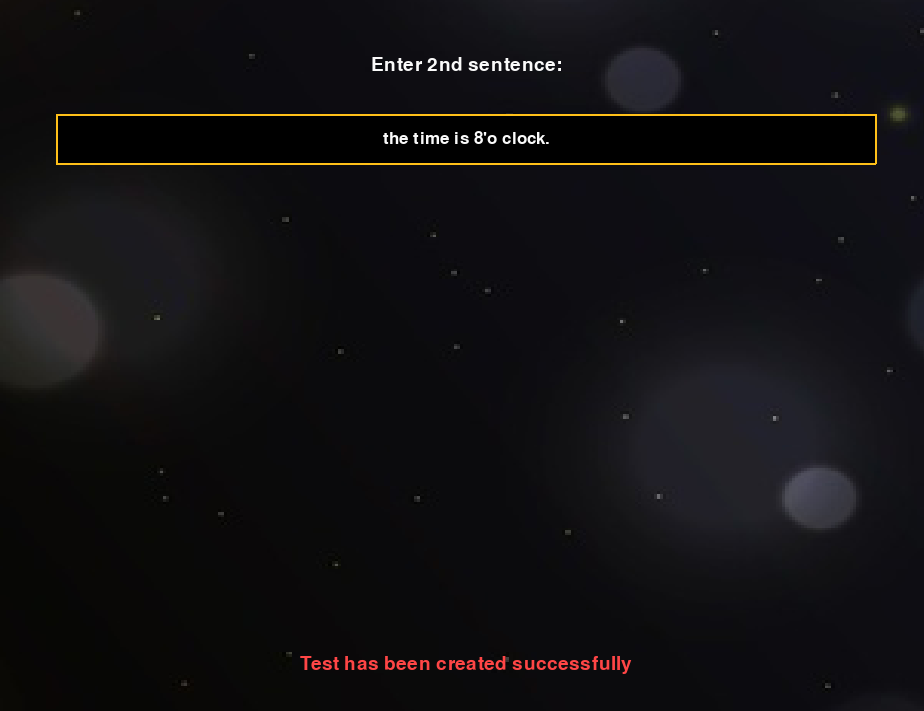
Create new test



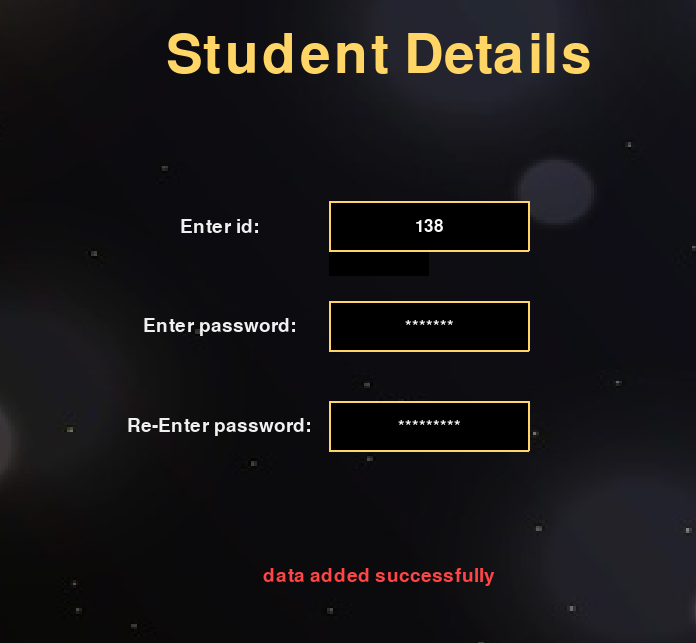
****



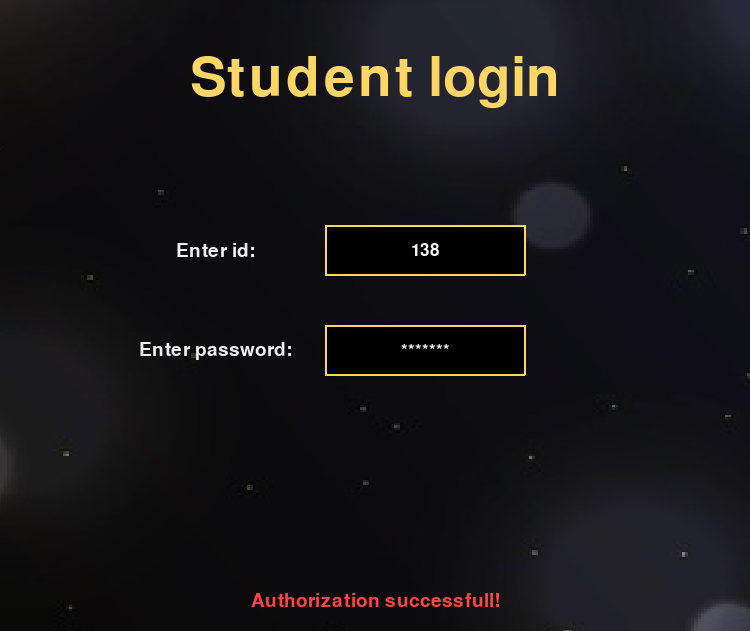


****

Create new student account

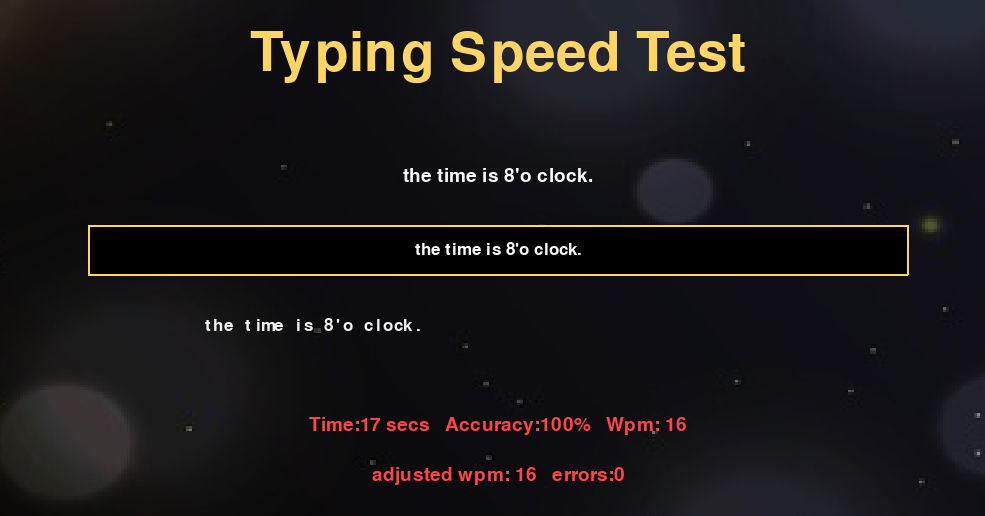


Join with test id

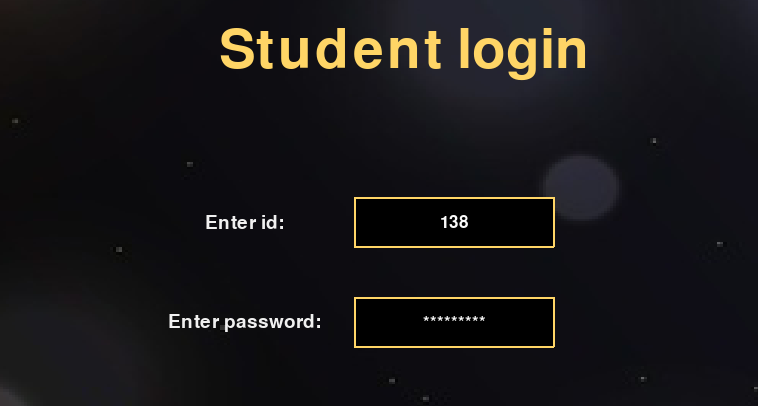


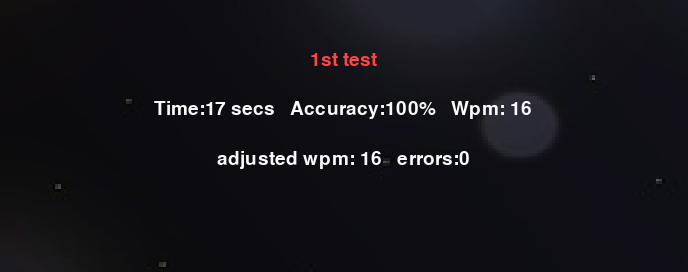




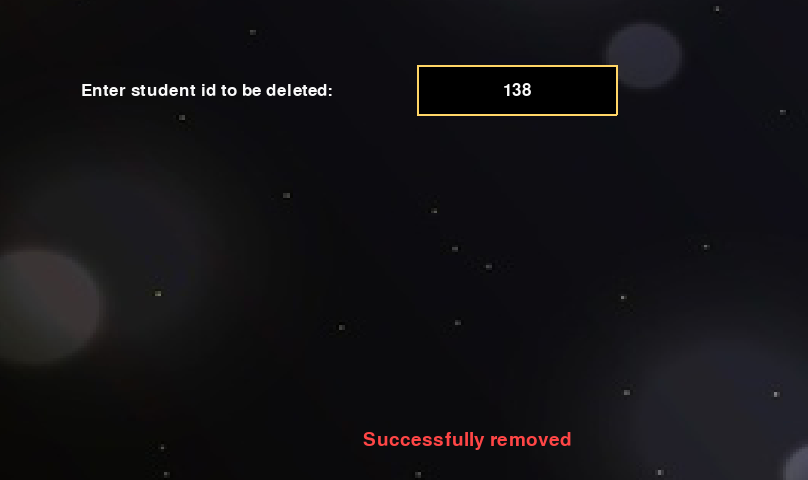


View stats

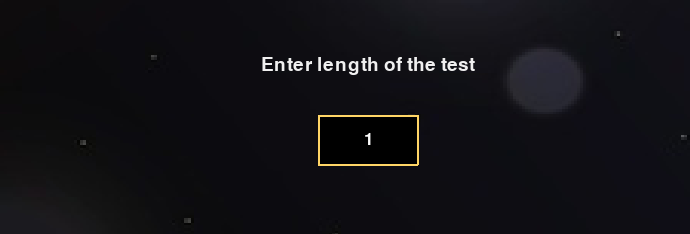


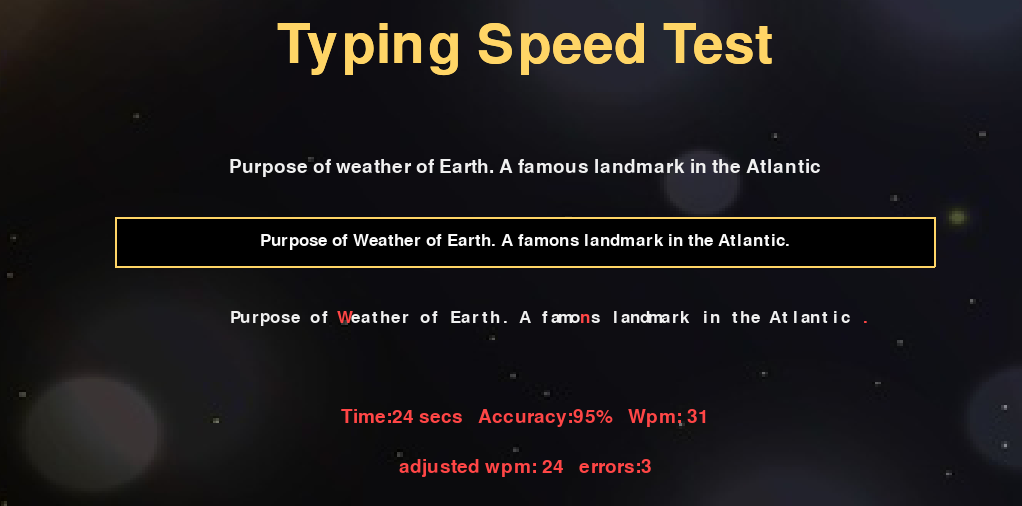


Delete student account



Start random test



****

**ADDITIONAL KNOWLEDGE ACQUIRED**

We have gained knowledge regarding many topics which weren’t the part of the curriculum until B.E - IInd Semester such as:

* Implementing GUI using pygame library.
* Randomizing sentences using essential generators.
* Calculating time at a specific block using time module.
* Designing Use Case Diagram and Activity Diagram.

**CONCLUSION AND FUTURE WORK**

The “Typing speed and accuracy test” project is a GUI based application which has a lot of features such as a creating test, storing accounts data and viewing stats. In future, we would like to make the improvements in graphics, more animations and add a graphical representation of data.

**REFERENCES**

Basic Python

1. Course covered during 2nd semester by Dr.Srinivas Chakravarthy
2. Ppts and handouts provided by the sir.

Pygame module

1. <https://www.geeksforgeeks.org/introduction-to-pygame/>

2. <https://pythonprogramming.net/pygame-python-3-part-1-intro/>

Randomizing sentences

1. <https://www.pythonpool.com/generate-random-sentence-in-python/>

Use cases and activity diagram

1. Hand-outs provided by SIREESHA ma’am.
2. <https://www.youtube.com/watch?v=zid-MVo7M-E>