

**TRAINING REPORT**  
**ON**  
**TYPING SPEED TEST**

Submitted to MAHARAJA RANJIT SINGH PUNJAB TECHNICAL  
UNIVERSITY in partial fulfillment of the requirement for the award of the  
degree of

**B. TECH**

**In**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted By**

**MOHIT, Roll. No. 190280077**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**GIANI ZAIL SINGH CAMPUS COLLEGE OF ENGINEERING &**  
**TECHNOLOGY, MRSPTU, BATHINDA-151001**

**DEC 2022**

**TRAINING REPORT**  
**ON**  
**TYPING SPEED TEST**

Submitted to MAHARAJA RANJIT SINGH PUNJAB TECHNICAL  
UNIVERSITY in partial fulfillment of the requirement for the award of the  
degree of

**B. TECH**

**In**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted By**

**MOHIT, Roll. No. 190280077**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**GIANI ZAIL SINGH CAMPUS COLLEGE OF ENGINEERING &**  
**TECHNOLOGY, MRSPTU, BATHINDA-151001**

**DEC 2022**

## **PREFACE**

Training is an integral part of B.Tech and each and every student has to undergo the training for **4 weeks** in a company.

This record is concerned about our practical training during the **7<sup>th</sup>** semester of our B.Tech. We have taken our Practical training in **INFOWIZ** During this training, we got to learn many new things about the industry and the current requirements of companies. This training proved to be a milestone in our knowledge of present industry. Every say and every moment was an experience in itself, an experience which theoretical study can't provide.

## **ACKNOWLEDGEMENT**

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced our thinking, behavior and acts during the course of study.

We express our sincere gratitude to ***Prof. Jyoti Rani*** worthy HOD, ***Er. Manpreet Kaur***, Training & Placement In-charge for providing us an opportunity to undergo summer training at INFOWIZ.

We are especially thankful to **Er. Manpreet Kaur** for his support, cooperation, and motivation provided to us during the project for constant inspiration, presence and blessings.

Lastly, we would like to thank the almighty and my parents for their moral support and our friends with whom we shared our day-to-day experience and received lots of suggestions that our quality of work.

**Mohit**

## **CANDIDATE'S DECLARATION**

I **Mohit**, Roll No. 190280077, B. Tech (Semester-VII) of the **Gaini Zail Singh Campus College of Engineering & Technology, Maharaja Ranjit Singh Punjab Technical University, Bathinda** hereby declare that the Training Report entitled “**PYTHON**” is an original work and data provided in the study is authentic to the best of my knowledge. This report has not been submitted to any other Institute for the award of any other degree.

**Mohit**

(Roll No. 190280077)

**Bathinda**

**Dec 2022**

H.O.: SCO 118 - 120, Sector 34 - A, CHANDIGARH  
B.O.: First Floor, Crown Tower, 100 Ft. Road, BATHINDA  
Web: [www.infowiz.co.in](http://www.infowiz.co.in) E-mail: [info@infowiz.co.in](mailto:info@infowiz.co.in)

# Certificate

No. INFOWIZ/2022/755

This is certified that Mr./Ms. MOHIT S/D/o. KRISHAN KUMAR

of GISCCET, Bathinda has successfully undergone Training Course PYTHON

From 16th July 2022 to 16th Aug. 2022. During the tenure of the above course, we found him/her a hardworking & innovative individual.

We wish him/her a very bright and prosperous future.

*Nishu*  
Technical Head



## Table of contents

CHAPTER 1: Introduction to Python .....	1
1.1    Introduction .....	1
1.2    Features in Python .....	1
1.3    Real world applications of Python .....	3
1.4    Why we choose Python .....	8
1.5    Python Advantages and Disadvantages .....	10
CHAPTER 2: Typing Speed Test .....	13
2.1    Project Description .....	13
2.2    System Specification .....	16
2.3    Game Screenshots .....	17
CHAPTER 3: Softwares Used .....	19
CHAPTER 4: Conclusion and future scope .....	20

# **CHAPTER 1: INTRODUCTION TO PYTHON**

## **1.1 Introduction to PYTHON**

Python is an interpreter, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding; make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

## **1.2 Features in Python**

There are many features in Python, some of which are discussed below –

### **1. Easy to code:**

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, JavaScript, Java, etc. It is very easy to code in python language and anybody can learn python basics in a few hours or days. It is also a developer friendly language.

### **2. Free and Open Source:**

Python language is freely available at the official website and you can download it from the official website.

Since it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.



### **3. Object-Oriented Language:**

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation, etc.

### **4. GUI Programming Support:**

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python.

PyQt5 is the most popular option for creating graphical apps with Python.

### **5. High-Level Language:**

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

### **6. Extensible feature:**

Python is an **Extensible** language. We can write us some Python code into C or C++ language and also we can compile that code in C/C++ language.

### **7. Python is Portable language:**

Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

### **8. Python is Integrated language:**

Python is also an Integrated language because we can easily integrated python with other languages like C, C++, etc.

### **9. Interpreted Language:**

Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called **byte code**.

## **10. Large Standard Library**

Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers, etc.

## **11. Dynamically Typed Language:**

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

## **1.3. Real-world Applications of Python**

### **1. Web Development**

When it comes to web development, Python should be your go-to tool. Why?

That's because Python offers numerous options for web development. For instance, you have Django, Pyramid, Flask, and Bottle for developing web frameworks and even advanced content management systems like Plone and Django CMS. These web frameworks are packed with standard libraries and modules which simplify tasks like content management, database interaction, and interfacing with internet protocols like HTTP, SMTP, XML, JSON, FTP, IMAP, and POP.

### **2. Game Development**

As we mentioned earlier, Python comes loaded with many useful extensions (libraries) that come in handy for the development of interactive games. For instance, libraries like PySoy (a 3D game engine that supports Python 3) and PyGame are two Python-based **libraries used widely for game development**. Python is the foundation for popular games like Battlefield 2, Frets on Fire, World of Tanks, Disney's Toontown Online, Vega Strike, and Civilization-IV.

Apart from game development, game designers can also use Python for developing tools to simplify specific actions such as level design or dialog tree creation, and even use those tools to export those tasks in formats that can be used by the primary game engine. Also, Python is used as a scripting language by many game engines.

### **3. Scientific and Numeric Applications**

Thanks to its massive library base, Python has become a crucial tool in scientific and numeric computing. In fact, Python provides the skeleton for applications that deal with computation and scientific data processing. Apps like FreeCAD (3D modeling software) and Abaqus (finite element method software) are coded in Python.

Some of the most useful Python packages for scientific and numeric computation include:

- SciPy (scientific numeric library)
- Pandas (data analytics library)
- IPython (command shell)
- Numeric Python (fundamental numeric package)
- Natural Language Toolkit (Mathematical and text analysis)

### **4. Artificial Intelligence and Machine Learning**

AI and ML models and projects are inherently different from traditional software models. When we talk about AI/ML projects, the tools and technologies used and the skillset required is totally different from those used in the development of conventional software projects. AI/ML applications require a language that is stable, secure, flexible, and is equipped with tools that can handle the various unique requirements of such projects. Python has all these qualities, and hence, it has become one of the most favoured languages of Data Science professionals.

Python's simplicity, consistency, platform independence, great collection of resourceful libraries, and an active community make it the perfect tool for developing AI and ML applications. Some of the best Python packages for AI and ML are:

- SciPy for advanced computing
- Pandas for general-purpose data analysis
- Seaborn for data visualization
- Keras, TensorFlow, and Scikit-learn for ML
- NumPy for high-performance scientific computing and data analysis

## 5. Desktop GUI

Python not only boasts of an English-like syntax, but it also features a modular architecture and the ability to work on multiple operating systems. These aspects, combined with its rich text processing tools, make Python an excellent choice for developing desktop-based GUI applications.

Python offers many GUI toolkits and frameworks that make desktop application development a breeze. PyQt, PyGtk, Kivy, Tkinter, WxPython, PyGUI, and PySide are some of the best Pythonbased GUI frameworks that allow developers to create highly functional Graphical User Interfaces (GUIs).

## 6. Software Development

Python packages and applications aim to simplify the process of software development. From developing complex applications that involve scientific and numeric computing to developing desktop and web applications, Python can do it all. This is the reason why **Software Developers use Python as a support language** for build control, testing, and management.

For instance, SCons is designed explicitly for build control, Buildbot and Apache Gump allow for automated continuous compilation and testing, and Roundup and Trac are great for bug tracking and project management.

Python also supports data analyzation and visualization, thereby further simplifying the process of creating custom solutions minus the extra effort and time investment.

## **7. Enterprise-level/Business Applications**

Enterprise-level software or business applications are strikingly different from standard applications, as in the former demands features like readability, extensibility, and scalability. Essentially, business applications are designed to fit the requirements of an organization rather than the needs of individual customers.

Thus, these applications must be capable of integrating with legacy systems like existing databases and non-web apps. Since business applications are developed, keeping in mind the custom requirements to cater to the specific needs of an organization's operating model, the entire development process becomes very complicated.

This is where Python can make a significant difference. Python high performance, scalability, flexibility, and readability are just the features required for developing fully-functional and efficient business applications.

## **8. Education programs and training courses**

If there's any beginner-friendly programming language, it is Python. We've said it many times before, and we're repeating it – Python has an extremely straightforward syntax that's similar to the English language. It has a short learning curve and hence, is an excellent choice for beginners. Python's easy learning curve and simplicity are the two main reasons why it is one of the most used programming languages in educational programs, both at beginner and advanced levels.

## **9. Language Development**

Over the years, Python's design and module architecture has been the inspiration behind the development of many new programming languages such as Boo, Swift, CoffeeScript, Cobra, and OCaml. All of these languages share numerous similarities with Python on grounds like object model, syntax, and indentation.

## **10. Operating Systems**

Yes, Python is the secret ingredient behind many operating systems as well, most popularly of Linux distributions. Linux-based Ubuntu's Ubiquity Installer and Fedora and Red Hat

Enterprise's Anaconda Installer are coded in Python. Even Gentoo Linux leverages Python Portage (package management system). Usually, Python is combined with the C programming language to design and develop operating systems.

### **11. Web Scraping Applications**

Python is a nifty tool for extracting voluminous amounts of data from websites and web pages. The pulled data is generally used in different real-world processes, including job listings, price comparison, R&D, etc.

### **12. Image Processing and Graphic Design Applications**

Alongside all the uses mentioned above, Python also finds a unique use case in image processing and graphic design applications. The programming language is used globally to design and build 2D imaging software like Inkscape, GIMP, Paint Shop Pro, and Scribus. Also, Python is used in several 3D animation packages such as Blender, Houdini, 3ds Max, Maya, Cinema 4D, and Lightwave, to name a few.

### **1.3. Why We Choose Python**

#### **1. Python is robust**

There is a good reason why Bank of America has chosen Python to power many of their critical systems. It's solid and powerful. Python has a relative small quantity of lines of code, which makes it less prone to issues, easier to debug, and more maintainable. The Securities Exchange Commission has sought to mandate Python as the language for a new "waterfall" program that would make Wall Street more transparent. Python can scale to solve complex problems, as illustrated by the fact that it powers most of YouTube and DropBox, not to mention Reddit, Quora, Disqus and FriendFeed. Even the mighty Google has made Python one of its official programming languages. It's also very fast.

#### **2. Python is flexible**

In 2007 YouTube migrated from PHP to Python for scalability purposes, citing that "Python enables flexibility". Python is used in a wide array of industries and for a long list of different usages, from websites and web applications to systems administration, voice over IP, and desktop apps. Python is also a staple of the Scientific community.

Because it wasn't originally created to answer a specific need, Python isn't driven by templates or specific APIs, and is therefore well-suited to rapid development of all kinds of applications. As a company focused on advanced web development, we really like this flexibility.

#### **3. Python is easy to learn and use**

"Python in particular emerges as a near ideal candidate for a first programming language", says John M. Zelle, in the Department of Mathematics, Computer Science, and Physics at Wartburg College in Iowa. We certainly agree with this as we find Python intuitive and fun. We don't have to look up references frequently, nor are we overwhelmed by the formalities of the language, like we would in Java or C++.

Python's simple and straight-forward syntax also encourages good programming habits, especially through its focus on white space indentation, which contributes to the development of neat looking code.

Finally, while PHP is notorious for the inconsistency in its naming methods, Python's naming convention is prevalent from module to module, so developers are less likely to make syntax errors. This means fewer bugs and faster development.

#### **4. Python reduces time to market**

Gartner estimates that 90% of enterprises are using open source software—including Perl, Python and Tcl—to build business-critical applications. That's because dynamic languages are an excellent solution for fast time-to-market for enterprise applications. Python makes it possible to get applications to market faster in part due to the fact that it has a huge standard library and is often referred to as coming with "batteries included". In addition, Python stays out of my way. Therefore, I can be more productive than if I was using Java/XML: the same task will require less code using Python.

#### **5. Python is free.**

Since Python is an open source programming language, we immediately reduce up-front project costs by leveraging Python in our development projects.



## **1.4. Python Advantages and Disadvantages**

### **Advantages of Python**

#### **1. Easy to Read, Learn and Write**

Python is a **high-level programming language** that has English-like syntax. This makes it easier to read and understand the code.

Python is really **easy to pick up and learn**, that is why a lot of people recommend Python to beginners. You need less lines of code to perform the same task as compared to other major languages like C/C++ and Java.

#### **2. Improved Productivity**

Python is a very **productive language**. Due to the simplicity of Python, developers can focus on solving the problem.

They don't need to spend too much time in understanding the syntax or behaviour of the programming language. You write less code and get more things done.

#### **3. Interpreted Language**

Python is an interpreted language which means that Python **directly executes the code** line by line. In case of any error, it stops further execution and reports back the error which has occurred.

Python shows only one error even if the program has multiple errors. This makes debugging easier.

#### **4. Dynamically Typed**

Python doesn't know the type of variable until we run the code. It automatically assigns the data type during execution. The programmer doesn't need to worry about declaring variables and their data types.

#### **5. Free and Open-Source**

Python comes under the **OSI approved** open-source license. This makes it **free to use** and distribute. You can download the source code, modify it and even distribute your version of Python. This is useful for organizations that want to modify some specific behaviour and use their version for development.

## 6. Vast Libraries Support

The standard library of Python is huge; you can find almost all the functions needed for your task. So, you don't have to depend on external libraries. But even if you do, a **Python package manager (pip)** makes things easier to import other great packages from the **Python package index (PyPi)**. It consists of over 200,000 packages.

## 7. Portability

In many languages like C/C++, you need to change your code to run the program on different platforms. That is not the same with Python. You only write once and run it anywhere. However, you should be careful not to include any **system-dependent features**.

## Disadvantages of Python

### 1. Slow Speed

We discussed above that Python is an interpreted language and dynamically-typed language. The line by line execution of code often leads to **slow execution**. The dynamic nature of Python is also responsible for the **slow speed** of Python because it has to do the extra work while executing code. So, Python is not used for purposes where speed is an important aspect of the project.

### 2. Not Memory Efficient

To provide simplicity to the developer, Python has to do a little trade-off. The Python programming language uses a **large amount of memory**. This can be a disadvantage while building applications when we prefer memory optimization.

Any doubts in Python advantages and disadvantages till now? Mention in the comment section. We will be happy to help you!!

### 3. Weak in Mobile Computing

Python is generally used in **server-side programming**. We don't get to see Python on the client side or mobile applications because of the following reasons. Python is **not memory efficient** and it has **slow processing power** as compared to other languages.

#### **4. Database Access**

Programming in Python is easy and stress-free. But when we are interacting with the database, it lacks behind. The Python's database access layer is primitive and underdeveloped in comparison to the popular technologies like **JDBC** and **ODBC**. Huge enterprises need smooth interaction of complex legacy data and Python is thus rarely used in enterprises.

#### **5. Runtime Errors**

As we know Python is a dynamically typed language so the data type of a variable can change anytime. A variable containing integer number may hold a string in the future, which can lead to **Runtime Errors**. Therefore, Python programmers need to perform thorough testing of the applications.

## **CHAPTER 2: TYPING SPEED TEST**

### **2.1 PROJECT DESCRIPTION:**

#### **About:**

In this Python project idea, we are going to build an exciting project through which you can **check** and even **improve** your typing speed. For a graphical user interface, we are going to use the **pygame** library which is used for working with graphics. We will draw the images and text to be displayed on the screen.

#### **Libraries used:**

For this project based on Python, we are using the pygame library. So we need to import the library along with some built-in modules of Python like time and random library.

Pip install pygame

#### **Game Class:**

The game class which will involve many functions responsible for starting the game, reset the game and few helper functions to perform calculations that are required for our project in Python.

We have initialized the width and height of the window, variables that are needed for calculation and then we initialized the pygame and loaded the images. The screen variable is the most important on which we will draw everything.

#### **draw\_text() method:**

The draw\_text() method of Game class is a helper function that will draw the text on the screen. The argument it takes is the screen, the message we want to draw, the y coordinate of the screen to position our text, the size of the font and color of the font. We will draw everything in the

center of the screen. After drawing anything on the screen, pygame requires you to update the screen.

### **get\_sentence() method:**

we have a list of sentences in our sentences.txt file? The get\_sentence() method will open up the file and return a random sentence from the list. We split the whole string with a newline character.

### **show\_results() method:**

The show\_results() method is where we **calculate the speed** of the user's typing. The time starts when the user clicks on the input box and when the user hits return key "Enter" then we perform the difference and calculate time in seconds.

To calculate accuracy, we did a little bit of math. We counted the correct typed characters by comparing input text with the display text which the user had to type.

*The formula for accuracy is:*

$(\text{correct characters}) * 100 / (\text{total characters in sentence})$

The WPM is the words per minute. A typical word consists of around 5 characters, so we calculate the words per minute by dividing the total number of words with five and then the result is again divided that with the total time it took in minutes. Since our total time was in seconds, we had to convert it into minutes by dividing total time with 60.

### **run() method:**

This is the main method of our class that will **handle all the events**. We call the reset\_game() method at the starting of this method which resets all the variables. Next, we run an infinite loop which will capture all the mouse and keyboard events. Then, we draw the heading and the input box on the screen.

We then use another loop that will look for the mouse and keyboard events. When the mouse button is pressed, we check the position of the mouse if it is on the input box then we start the time and set the active to True. If it is on the reset button, then we reset the game.

When the active is True and typing has not ended then we look for keyboard events. If the user presses any key, then we need to update the message on our input box. The enter key will end typing and we will calculate the scores to display it. Another event of a backspace is used to trim the input text by removing the last character.

### **reset\_game() method:**

The reset\_game() method **resets all variables** so that we can start testing our typing speed again. We also select a random sentence by calling the get\_sentence() method. In the end, we have closed the class definition and created the object of Game class to run the program.

## **2.2 SYSTEM SPECIFICATIONS:**

### **Hardware Requirements:**

Processor	– Pentium 4
RAM	– 1 GB
Hard Disk	– 40GB
Mouse	– Standard Mouse
Keyboard	–Standard Keyboard
Processor Speed	– 2.4GHZ

### **Software Requirements:**

Operating System	– Microsoft Windows XP With Service Pack 2
Coding	– Pycharm 2022.3

### **Display Mode:**

Color Quality	– Highest[32 bit]
Screen Resolution	– 1024 by 768 Pixels

## 2.3 PROJECT SCREENSHOTS:

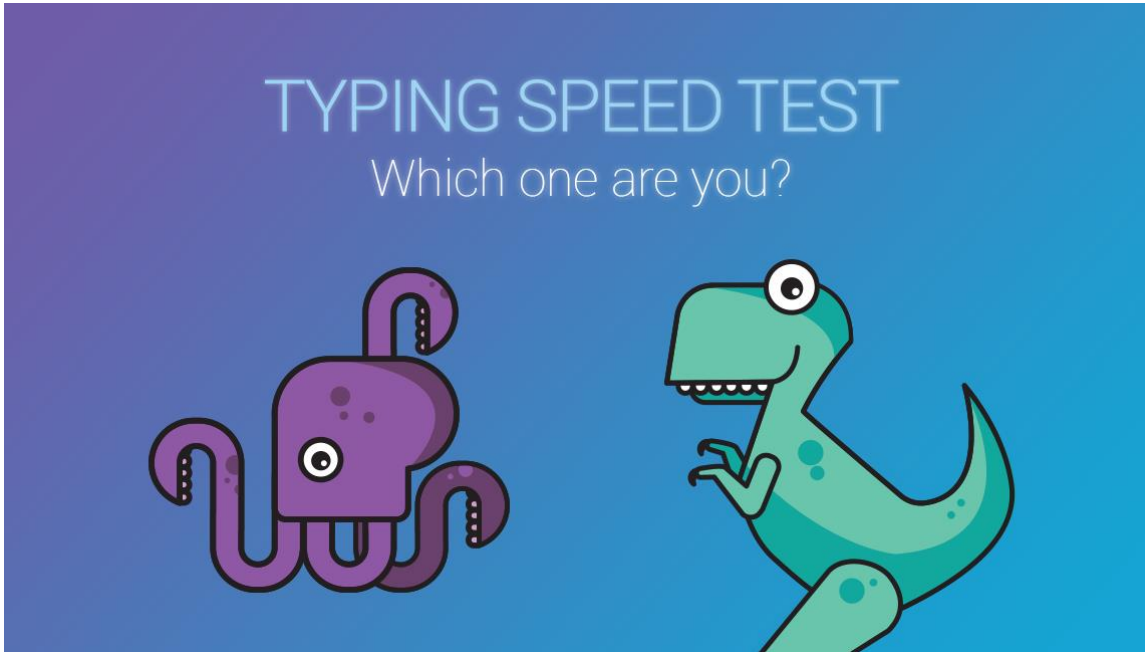


Figure 2.3.1 First page of typing speed test app

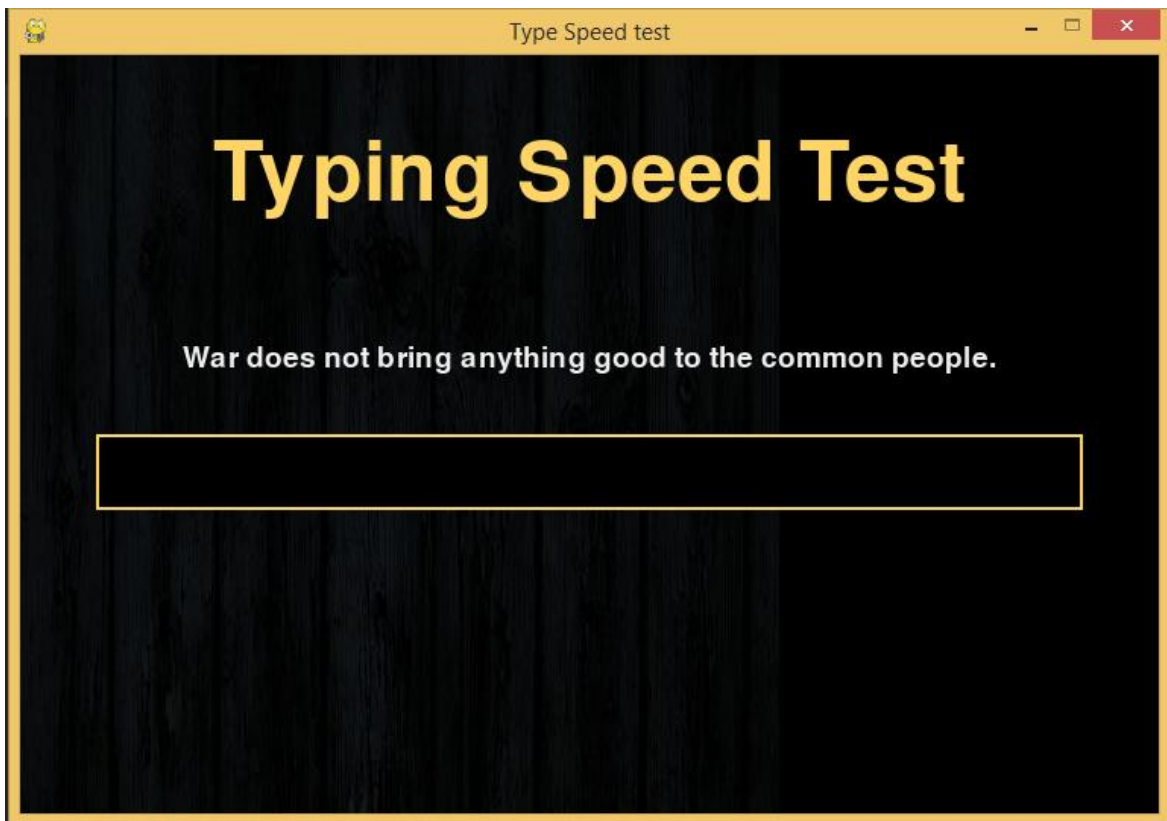


Figure 2.3.2 Text- box for typing





Figure 2.3.3 Results of the test

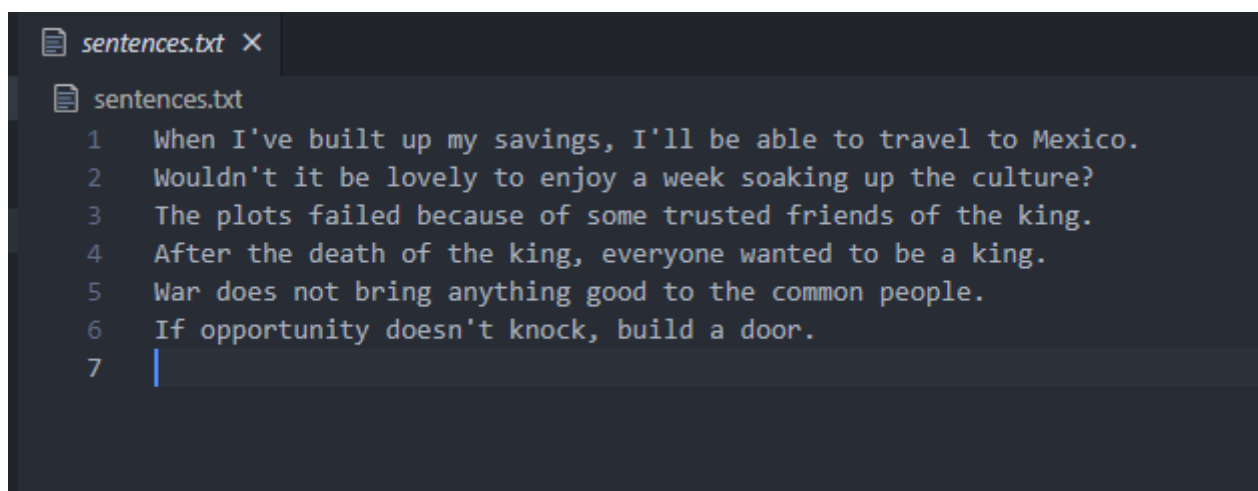


Figure 2.3.2 Sentences stored in the txt file of app

## **CHAPTER 3: Software used**

### **PyCharm:**

PyCharm is an Integrated Development Environment (IDE) used for programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django. PyCharm is developed by the Czech company JetBrains.

It is cross-platform working on Windows, Mac OS X and Linux. PyCharm has a Professional Edition, released under a proprietary license and a Community Edition released under the Apache License. PyCharm Community Edition is less extensive than the Professional Edition.

Features:

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: including rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask
- Integrated Python debugger
- Integrated unit testing, with line-by-line coverage
- Google App Engine Python development
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with changelists and merge.

## **CHAPTER 4: Conclusion and Future Scope**

To conclude I created a Typing Speed Tester that consists of basic feature like telling the users their typing speed which makes this fun to play. The development process was not less than any adventure journey we're on through out these couple of weeks. We hope that this app will be liked by others. We'll keep adding new features to the app for making it more attractive and user friendly. We'll be continuously updating this app. Our focus will be on adding the essential features first like:

- Hositng this app publically on the internet.
- Adding more sentences.
- Creating different levels to make it more fun.
- Providing detailed analysis of user's performance.
- Including typing tips and lessons.

and much more...