PROJECT 1 :TYPING SPEED CALCULATOR

NAME. : Vinayak Shivaling Vijayanagar

USN : 1EP21AD057

G MAIL :

rani.vini.viddi.sakshi@gmail.com

ACKNOWLEDGEMENT

While I was making this project , a lot of information that I found helped me in chapter and I am glad that I was able to complete this project and was able to understand many things.

It gave me an immense pleasure while doing this project because it was not just a project but , a source to learn not just about chapter but also I inculcated many qualities like responsibility , punctuality , confidence and what not.

The journey of making this project was so nice and in all this my teachers who supported me all the time , cleared my doubts and the parents support also played a big role and my friends too helped me . I thank to all of them and wish that they keep supporting me like this.

Doing work on time is something everyone needs to learn and through this project I have improved my timing and also it made my thinking skills better .

A project is a bridge between theoretical and practical learning and with this thinking I worked on the project and made it successful due to timely support and efforts of all who helped me.

Once again thank you Sir/Ma'am to give me this project and to make me learn so many things .I have no more valuable words to express my thanks , but my heart is still full of favor received from every person

ABOUT THE COMPANY:

Varcons Technologies Pvt Ltd is a digital service provider that aims to provide software, designing and marketing solutions to individuals and businesses. At VCT, we believe that service and quality is the key to success.

We provide all kinds of technological and designing solutions from Billing Software to Web Designs or any custom demand that you may have. Experience the service like none other!

Some of our services include:

Development - We develop responsive, functional and super-fast websites. We keep User Experience in mind while creating websites. A website should load quickly and should be accessible even on a small view-port and slow internet connection, We develop sophisticated customizable softwares using Java and other programming Languages as per the clients needs'

Mobile Application - We offer a wide range of professional android, iOS & Hybrid app development services for our global clients, from a start up to a large enterprise.

Design - We offer professional Graphic design, Brochure design & Logo design. We are experts in crafting visual content to convey the right message to the customers.

Consultancy - We are here to provide you with expert advice on your design and development requirement.

Videos - We create a polished professional video that impresses your audience

INDEX

A BRIEF OVERVIEW ON THE PROJECT:

This Java project is a general-purpose, concurrent, class-based, object- oriented computer programming language that is specifically designed to have as few implementation dependencies as possible and it helps to learn how to define and call the function in code.

It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another and from doing this project we get know about class must allow the objects to cooperate during the execution

2.ADVANTAGES AND CHALLENGES WHILE COMPLETING THE PROJECT.

This project has advantage of giving the user a understanding of his /her speed in typing the words and characters, also mentioning the correct accuracy of typed words. THE TYPING SPEED CACULATOR done has levels to it such as easy , medium , hard indicating the user speeds in various conditions given the randomness in words and sentences given helps the user to achieve more accuracy through his /her practice in the typing speed. Finally the WPM , CPM and accuracy is output for various levels user completes.

On the other hand the code of various levels that was to be done was challenge along with the time calculated for each input and the usual problems like use of inbuilt functions and

minor problems in methods were faced often but overcome in final result.

SOFTWARE AND HARDWARE REQUIRMENTS FOR JAVA PROJECT :

Hardware Requirement for Java :

Minimum hardware requirement to download Java on your Windows operating system as follows:

- Minimum Windows 95 software
- IBM-compatible 486 system
- Hard Drive and Minimum of 8 MB memory
- A CD-ROM drive
- Mouse, keyboard and sound card

Software requirement for Java :

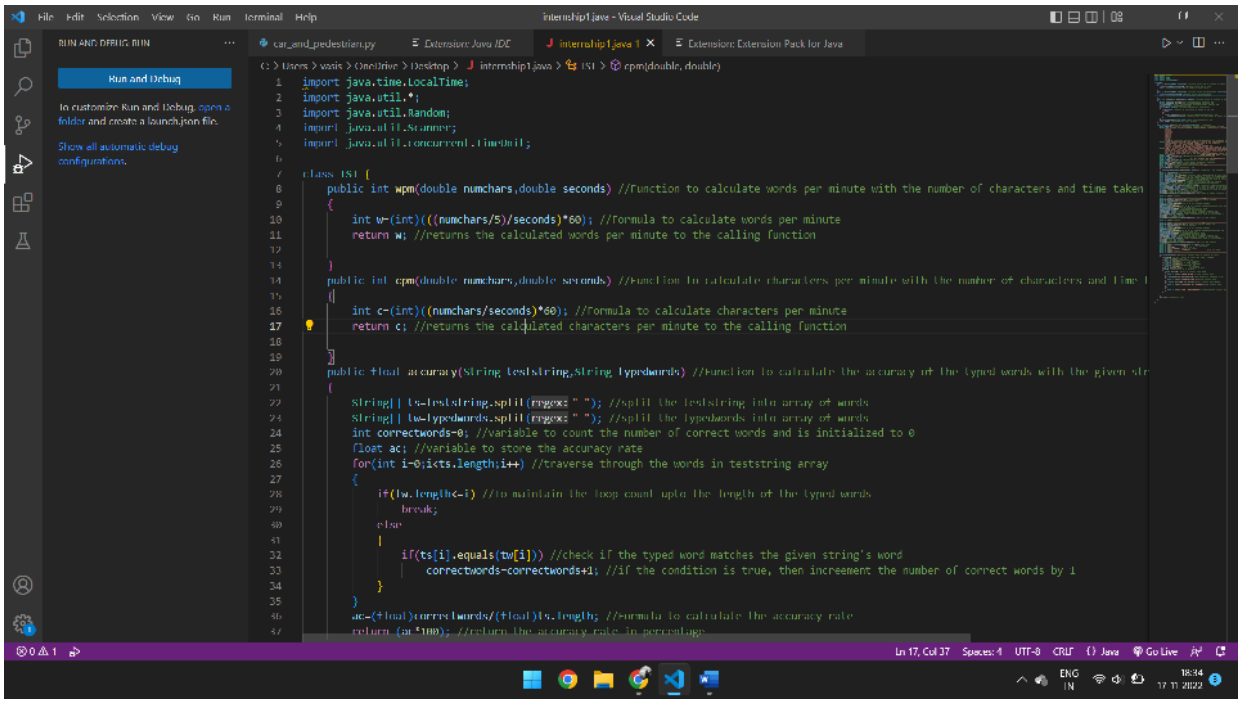
Nowadays, Java is supported by almost every operating systems. Whether it is a Windows, Macintosh and Unix all supports the Java application development. So you can download any of the operating system on your personal computer. Here are the minimum requirement.

The software require for project includes package necessary to successfully compile and build the program.

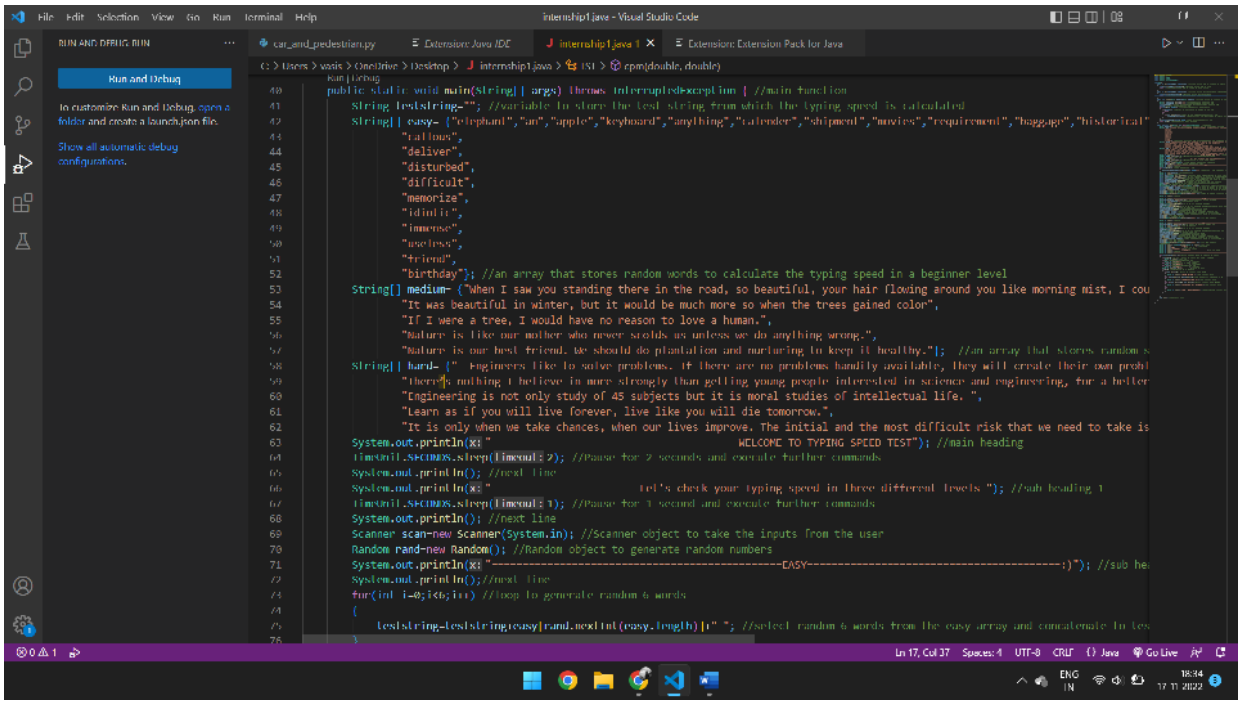
- Operating System
- Java SDK or JRE 1.6 or higher
- Java Servlet Container (Free Servlet Container available)
- Supported Database and library that supports the database connection with Java.

TYPING SPEED CALCULATOR

IMPLEMENTATION OF THE PROJECT with Screenshots:

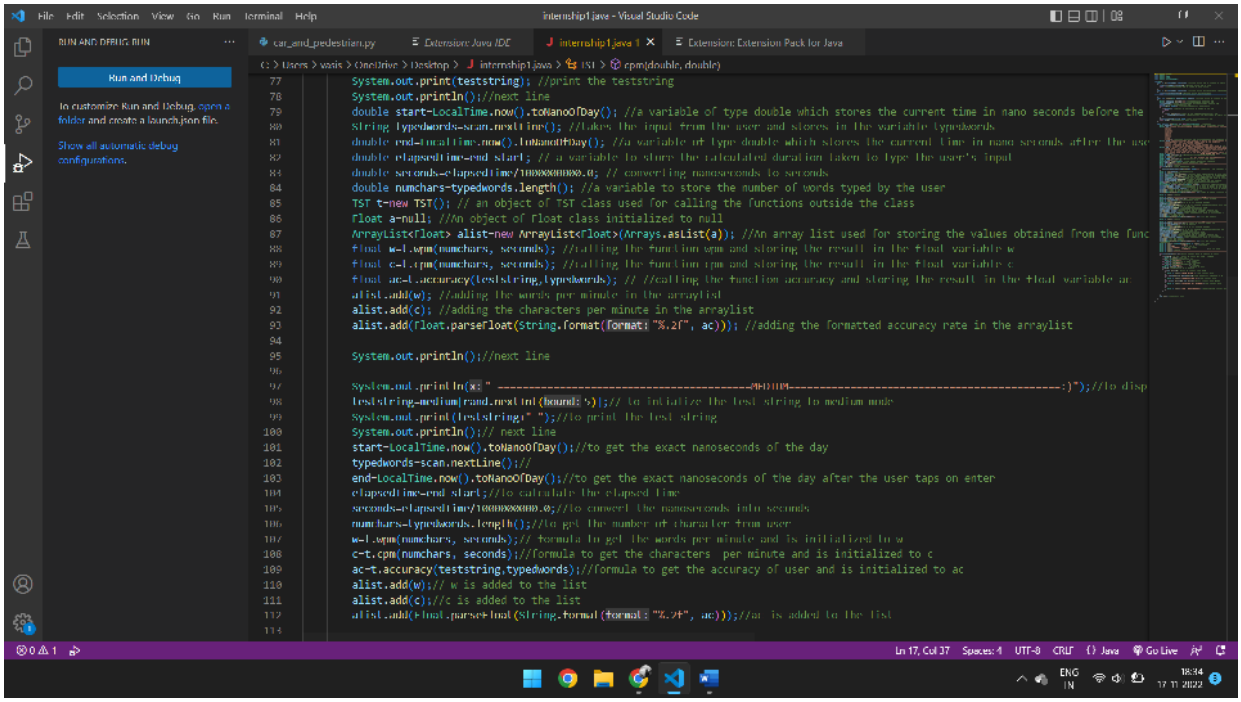


```java
import java.time.LocalTime;
import java.util.*;
import java.util.Random;
import java.util.Scanner;
import java.util.concurrent.TimeUnit;

class IST {
    public int wpm(double numchars,double seconds) //function to calculate words per minute with the number of characters and time taken
    {
        int w=(int)(((numchars/5)/seconds)*60); //formula to calculate words per minute
        return w; //returns the calculated words per minute to the calling function

    }
    public int cpm(double numchars,double seconds) //function to calculate characters per minute with the number of characters and time t
    {
        int c=(int)((numchars/seconds)*60); //formula to calculate characters per minute
        return c; //returns the calculated characters per minute to the calling function

    }
    public float accuracy(String teststring,String typedwords) //function to calculate the accuracy of the typed words with the given str
    {
        String[] ts=teststring.split(regex: " "); //split the teststring into array of words
        String[] tw=typedwords.split(regex: " "); //split the typedwords into array of words
        int correctwords=0; //variable to count the number of correct words and is initialized to 0
        float ac; //variable to store the accuracy rate
        for(int i=0;i<ts.length;i++) //traverse through the words in teststring array
        {
            if(tw.length<=i) //to maintain the loop count upto the length of the typed words
                break;
            else
            {
                if(ts[i].equals(tw[i])) //check if the typed word matches the given string's word
                    correctwords=correctwords+1; //if the condition is true, then increement the number of correct words by 1
            }
        }
        ac=(float)correctwords/(float)ts.length; //formula to calculate the accuracy rate
        return (ac*100); //return the accuracy rate in percentage
```

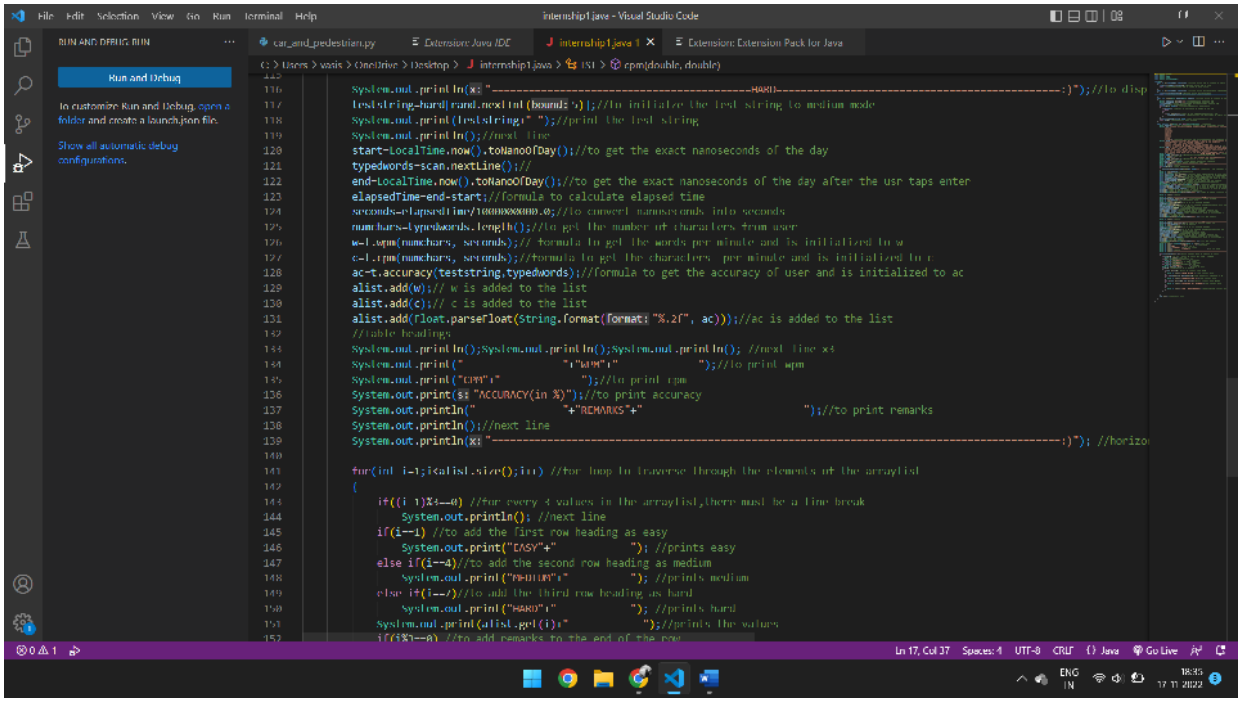


```java
    public static void main(String[] args) throws InterruptedException { //main function
        String teststring=""; //variable to store the test string from which the typing speed is calculated
        String[] easy= {"elephant","an","apple","keyboard","anything","calender","shipment","movies","requirement","baggage","historical"
                "rallous",
                "deliver",
                "disturbed",
                "difficult",
                "memorize",
                "idiotic",
                "immense",
                "useless",
                "friend",
                "birthday"}; //an array that stores random words to calculate the typing speed in a beginner level
        String[] medium= {"When I saw you standing there in the road, so beautiful, your hair flowing around you like morning mist, I cou
                "It was beautiful in winter, but it would be much more so when the trees gained color",
                "If I were a tree, I would have no reason to love a human.",
                "Nature is like our mother who never scolds us unless we do anything wrong.",
                "Nature is our best friend. We should do plantation and nurturing to keep it healthy."};  //an array that stores random s
        String[] hard= {" Engineers like to solve problems. If there are no problems handily available, they will create their own probl
                "there's nothing I believe in more strongly than getting young people interested in science and engineering, for a better
                "Engineering is not only study of 45 subjects but it is moral studies of intellectual life. ",
                "Learn as if you will live forever, live like you will die tomorrow.",
                "It is only when we take chances, when our lives improve. The initial and the most difficult risk that we need to take is
        System.out.println(x: "                                                WELCOME TO TYPING SPEED TEST"); //main heading
        TimeUnit.SECONDS.sleep(timeout: 2); //Pause for 2 seconds and execute further commands
        System.out.println(); //next line
        System.out.println(x: "                                    Let's check your typing speed in three different levels "); //sub heading 1
        TimeUnit.SECONDS.sleep(timeout: 1); //Pause for 1 second and execute further commands
        System.out.println(); //next line
        Scanner scan=new Scanner(System.in); //Scanner object to take the inputs from the user
        Random rand=new Random(); //Random object to generate random numbers
        System.out.println(x: "-----------------------------------------EASY-----------------------------------------:)"); //sub he
        System.out.println();//next line
        for(int i=0;i<6;i++) //loop to generate random 6 words
        {
            teststring=teststring+easy[rand.nextInt(easy.length)]+" "; //select random 6 words from the easy array and concatenate to tes
        }
```

TYPING SPEED CALCULATOR

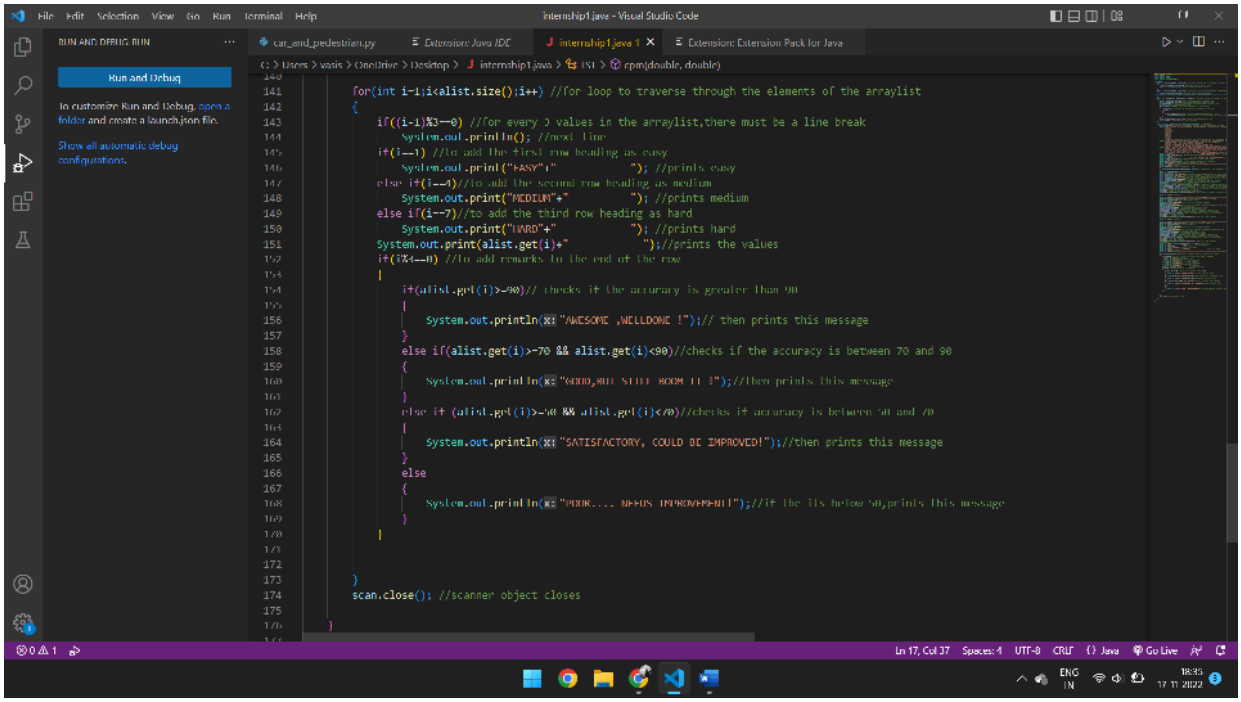


```
        System.out.print(teststring); //print the teststring
        System.out.println();//next line
        double start=LocalTime.now().toNanoOfDay(); //a variable of type double which stores the current time in nano seconds before the
        String typedwords=scan.nextLine(); //takes the input from the user and stores in the variable typedwords
        double end=LocalTime.now().toNanoOfDay(); //a variable of type double which stores the current time in nano seconds after the use
        double elapsedTime=end start; // a variable to store the calculated duration taken to type the user's input
        double seconds=elapsedTime/1000000000.0; // converting nanoseconds to seconds
        double numchars=typedwords.length(); //a variable to store the number of words typed by the user
        TST t=new TST(); // an object of TST class used for calling the functions outside the class
        Float a=null; //An object of Float class initialized to null
        ArrayList<Float> alist=new ArrayList<Float>(Arrays.asList(a)); //An array list used for storing the values obtained from the func
        float w=t.wpm(numchars, seconds); //calling the function wpm and storing the result in the float variable w
        float c=t.cpm(numchars, seconds); //calling the function cpm and storing the result in the float variable c
        float ac=t.accuracy(teststring,typedwords); // //calling the function accuracy and storing the result in the float variable ac
        alist.add(w); //adding the words per minute in the arraylist
        alist.add(c); //adding the characters per minute in the arraylist
        alist.add(Float.parseFloat(String.format("%.2f", ac))); //adding the formatted accuracy rate in the arraylist

        System.out.println();//next line

        System.out.println(" ----------MEDIUM----------:)");//to disp
        teststring=medium[rand.nextInt(5)];// to initialize the test string to medium mode
        System.out.print(teststring+" ");//to print the test string
        System.out.println();// next line
        start=LocalTime.now().toNanoOfDay();//to get the exact nanoseconds of the day
        typedwords=scan.nextLine();//
        end=LocalTime.now().toNanoOfDay();//to get the exact nanoseconds of the day after the user taps on enter
        elapsedTime=end start;//to calculate the elapsed time
        seconds=elapsedTime/1000000000.0;//to convert the nanoseconds into seconds
        numchars=typedwords.length();//to get the number of character from user
        w=t.wpm(numchars, seconds);// formula to get the words per minute and is initialized to w
        c=t.cpm(numchars, seconds);//formula to get the characters  per minute and is initialized to c
        ac=t.accuracy(teststring,typedwords);//formula to get the accuracy of user and is initialized to ac
        alist.add(w);// w is added to the list
        alist.add(c);//c is added to the list
        alist.add(Float.parseFloat(String.format("%.2f", ac)));//ac  is added to the list
```



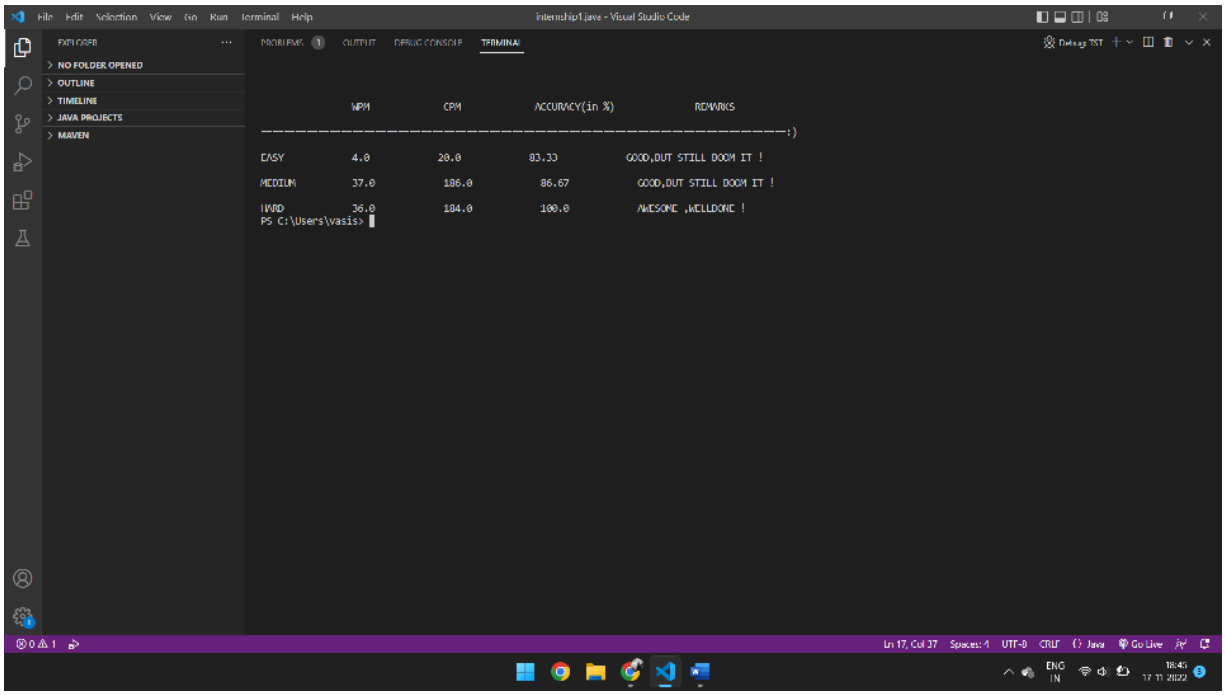
```
        System.out.print(" ----------HARD----------:)");//to disp
        teststring=hard[rand.nextInt(5)];//to initialize the test string to medium mode
        System.out.print(teststring+" ");//print the test string
        System.out.println();//next line
        start=LocalTime.now().toNanoOfDay();//to get the exact nanoseconds of the day
        typedwords=scan.nextLine();//
        end=LocalTime.now().toNanoOfDay();//to get the exact nanoseconds of the day after the usr taps enter
        elapsedTime=end start;//formula to calculate elapsed time
        seconds=elapsedTime/1000000000.0;//to convert nanoseconds into seconds
        numchars=typedwords.length();//to get the number of characters from user
        w=t.wpm(numchars, seconds);// formula to get the words per minute and is initialized to w
        c=t.cpm(numchars, seconds);//formula to get the characters  per minute and is initialized to c
        ac=t.accuracy(teststring,typedwords);//formula to get the accuracy of user and is initialized to ac
        alist.add(w);// w is added to the list
        alist.add(c);// c is added to the list
        alist.add(Float.parseFloat(String.format("%.2f", ac)));//ac is added to the list
        //table headings
        System.out.println();System.out.println();System.out.println(); //next line x4
        System.out.print("          "+"WPM"+"          ");//to print wpm
        System.out.print("CPM"+"          ");//to print cpm
        System.out.print("ACCURACY(in %)");//to print accuracy
        System.out.println("          "+"REMARKS"+"          ");//to print remarks
        System.out.println();//next line
        System.out.println(" ----------:)"); //horizo

        for(int i=1;i<alist.size();i++) //for loop to traverse through the elements of the arraylist
        {
            if((i-1)%4==0) //for every 4 values in the arraylist,there must be a line break
                System.out.println(); //next line
            if(i==1) //to add the first row heading as easy
                System.out.print("EASY"+"          "); //prints easy
            else if(i==4)//to add the second row heading as medium
                System.out.print("MEDIUM"+"          "); //prints medium
            else if(i==7)//to add the third row heading as hard
                System.out.print("HARD"+"          "); //prints hard
            System.out.print(alist.get(i)+"          ");//prints the values
            if(i%3==0) //to add remarks to the end of the row
```

TYPING SPEED CALCULATOR



```
        for(int i=1;i<alist.size();i++) //for loop to traverse through the elements of the arraylist
        {
            if((i-1)%3==0) //for every 3 values in the arraylist,there must be a line break
                System.out.println(); //next line
            if(i==1) //to add the first row heading as easy
                System.out.print("EASY"+"            "); //prints easy
            else if(i==4)//to add the second row heading as medium
                System.out.print("MEDIUM"+"          "); //prints medium
            else if(i==7)//to add the third row heading as hard
                System.out.print("HARD"+"            "); //prints hard
            System.out.print(alist.get(i)+"                "};//prints the values
            if(i%3==0) //to add remarks to the end of the row
            {
                if(alist.get(i)>=90)// checks if the accuracy is greater than 90
                {
                    System.out.println("AWESOME ,WELLDONE !");// then prints this message
                }
                else if(alist.get(i)>=70 && alist.get(i)<90)//checks if the accuracy is between 70 and 90
                {
                    System.out.println("GOOD,BUT STILL BOOM IT !");//then prints this message
                }
                else if (alist.get(i)>=50 && alist.get(i)<70)//checks if accuracy is between 50 and 70
                {
                    System.out.println("SATISFACTORY, COULD BE IMPROVED!");//then prints this message
                }
                else
                {
                    System.out.println("POOR.... NEEDS IMPROVEMENT");//if the its below 50,prints this message
                }
            }


        }
        scan.close(); //scanner object closes
    }
```

OUTPUT:



```
            WPM             CPM             ACCURACY(in %)          REMARKS
-------------------------------------------------------------------------:)

EASY        4.0             20.0            83.33           GOOD,BUT STILL BOOM IT !

MEDIUM      37.0            186.0           86.67             GOOD,BUT STILL BOOM IT !

HARD        36.0            184.0           100.0             AWESOME ,WELLDONE !
PS C:\Users\vasis>
```

CONCLUSION:

The successful implementation of the TYPING SPEED CALCULATOR is done as per following the use of WPM(WORDS PER MINUTE) , CPM (CHARACTERS PER MINUTE), ACCURACY of the typed words. The project uses the efficient use of java and advanced java concepts to develop this project into final successful result.

BIBLIOGRAPHY:

- "Core Java™, Volume I--Fundamentals (8th Edition) " , by Cay S. Horstmann, Prentice Hall; 8
edition (April 18, 2008).

- "Effective Java (2nd Edition)" , by Addison-Wesley; 2 edition (May 28, 2008) .

- "Java The Complete Reference, 8th Edition", McGraw-Hill Osborne Media; 8 edition (June 22, 2011).

- A Programmer's Guide to Java SCJP Certification: A Comprehensive Primer (3rd Edition), Addison-Wesley Professional; 3 edition (December 29, 2008).

- "More Java Pitfalls: 50 New Time-Saving Solutions and Workarounds" , by Michael C. Daconta (Author), Kevin T. Smith (Author), Donald Avondolio (Author), W. Clay Richardson (Author), Wiley; 1 edition (February 3, 2003).

- "Head First Servlets and JSP: Passing the Sun Certified Web Component Developer Exam ", by Bryan Basham (Author), Kathy Sierra (Author), Bert Bates (Author), O'Reilly Media; Second Edition edition (April 1, 2008).

- "Head First Design Patterns", Elisabeth Freeman (Author), Eric Freeman (Author), Bert Bates (Author), Kathy Sierra (Author), Elisabeth Robson (Author), O'Reilly Media; 1 edition (November 1, 2004)

- "EJB 3 in Action", Debu Panda (Author), Reza Rahman (Author), Derek Lane (Author),
Manning Publications; 1 edition (April 16, 2007).

- "JBoss: A Developer's Notebook" , Norman Richards (Author), Sam Griffith (Author), O'Reilly

- Enterprise Java™ Security: Building Secure J2EE™ Applications, Marco Pistoia (Author), Nataraj Nagaratnam (Author), Larry Koved (Author), Anthony Nadalin (Author), Addison- Wesley

Professional; 1 edition (February 27, 2004).

- "Java EE 6 Development with NetBeans 7", Author: David Heffelbinger, Published: June 2011, PACKT Publishing, UK. NetBeans IDE 7.0 Cookbook, Author: Rhawi Dantas, Published: May 2011, PACKT Publishing, UK

- "Art of Java Web Development: Struts, Tapestry, Commons, Velocity, JUnit, Axis, Cocoon, InternetBeans, WebWork", Neal Ford (Author), Manning Publications (November 1, 2003).

- "Struts 2 in Action", Don Brown (Author), Chad Michael Davis (Author), Scott Stanlick (Author), Manning Publications; 1 edition (May 1, 2008).

- "Hibernate in Action (In Action series)", Christian Bauer (Author), Gavin King (Author), Manning Publications (August 1, 2004) "Web Services Essentials (O'Reilly XML)", Ethan Cerami (Author), O'Reilly Media (February 2002) .

- "Object-Oriented Software Construction (Book/CD-ROM) (2nd Edition)" ,Bertrand Meyer (Author), Prentice Hall; 2nd edition (March 21, 2000).

- Java Black Book, Steve Holzner (Author), Steven Holzner (Author), Paraglyph Press; Second Edition edition (July 1, 2002)

TYPING SPEED CALCULATOR

TYPING SPEED CALCULATOR