**SPL-1 Project Report**

**Typing Master: An Interactive Typing Tutor and Typing Game**

Submitted by

***Mohammad Ismail Hossain***

**BSSE Roll No.: 1433**

**BSSE Session: 2021-22**

Submitted to

***Dr. Emon Kumar Dey***

**Associate Professor**

**Institute of Information Technology**

Signature of Supervisor: ............................

Date: .......................................

****

**Institute of Information Technology**

**University of Dhaka**

**17 - 12 - 2023**

**Table of Contents**

**1. Introduction** …………………………………………………………………………………………. 1

**2. Background Study** ..................................................................... 1

2.1 Necessity of Typing Tutor Software …………………………………………………. 1

2.2 Objectives of this project …………………………………………………………………. 2

**3. Description** ................................................................................ 2

3.1 Login System ……………………………………………………………………………………. 2

3.2 Typing Tutorial ………………………………………………………………………………….. 2

3.3 Typing Practice …………………………………………………………………………………. 3

3.4 Typing Game .......................................................................... 3

3.5 Performance History ................................................................ 3

**4. Implementation and Testing** ....................................................... 3

# 4.1 Libraries and header files .......................................................... 3

# 4.2 Function Definition ................................................................... 4

4.3. File I/O operation .................................................................... 5

# 4.4 Bangla text I/O operation .......................................................... 5

# 4.5 Bangla text to Hex Unicode convert ............................................ 6

# 4.6 Unicode hex value to Bangla text convert .................................... 6

# 4.7 Typing Time calculation ............................................................ 7

# 4.8 Typing Speed (WPM) Calculate .................................................. 7

# 4.9 Accuracy Calculate ................................................................... 7

# 4.10 Difficult keys analysis using histogram ...................................... 8

# 4.11 Edit Distance Algorithm .......................................................... 8

# 4.12 Random word generate ........................................................... 9

# 4.13 Cursor move .......................................................................... 10

# 4.14 Login System ......................................................................... 10

# 4.15 Performance history save ......................................................... 11

# 5. User Interface ............................................................................ 12

5.1 Login System .......................................................................... 12

# 5.2 Language Selection Page .......................................................... 12

# 5.3 Home Page ............................................................................. 13

# 5.4 Tutorial Page ........................................................................... 13

# 5.5 Typing Practice Page ................................................................ 14

# 5.6 Typing Game Page ................................................................... 15

# 5.7 Performance History ................................................................. 16

# 6. Challenges Faced ........................................................................ 16

6.1 Learning C++ .......................................................................... 16

6.2 Handling a Large Project ........................................................... 16

6.3 File Handling Operations ........................................................... 17

6.4 Bangla Language Support ......................................................... 17

6.5 Creating timer and time calculation ............................................ 17

**7. Conclusion** .................................................................................. 17

**Reference** ........................................................................................ 18

**List of Figures**

1. File writing process ……………………………………………………………………………………... 5

2. File read process ……………………………………………………………………………………….... 5

3. Bangla Unicode text read form file and output ………………………………………….. 6

4. Bangla text to Unicode conversion ……………………………………………………………… 6

5. Hexadecimal Unicode value to Bangla text conversion ……………………………… 7

6. Calculate typing time duration ……………………………………………………………………. 7

7. Calculate typing speed in WPM ……………………………………………………………………. 7

8. Histogram drawing based on the error frequency ……………………………………… 8

9. Score giving process in game part ……………………………………………………………… 8

10. Edit Distance DP Algorithm ……………………………………………………………………….. 9

11. Random word generate ……………………………………………………………………………… 9

12. Cursor moving in console screen ………………………………………………………………. 10

13: Caesar Cipher Algorithm implementation............................................. 10

14. Sign Up process …………………………………………………………............................. 11

15. Login process …………………………………………….……………………………………………….. 11

16. Performance history save to text file .................................................. 11

17. Welcome page ................................................................................ 12

18. Sign Up page …………………………………………….………………………………............... 12

19. Login page …………………………………………….…………………………………….............. 12

20. Language Selection page ................................................................. 13

21. Home page .................................................................................... 13

22. Tutorial page …………………………………………….……………………………………........... 13

23. Practice type selection page ............................................................. 14

24. Practice page …………………………………………….…………………………………….......... 14

25. Practice result analysis …………………………………………….……………………………….. 15

26. Fast Typer Game (English) .............................................................. 15

27. Fast Typer Game (Bangla) …………………………………………….…………………………. 16

28. Performance history ........................................................................ 16

**1**

**1.** **Introduction**

The Typing Tutor and Typing Game Project is an interactive application designed to enhance typing skills while making the learning process interactive and effective. This application includes typing tutorial, typing practice exercises, typing games, detailed performance analysis. This application support both Bangla and English language. By integrating these functionalities, the project aims to provide a platform suitable for beginners looking to improve their typing skill. The incorporation of Bangla typing ensures useability for Bengali language users.

This project is mainly divided into 3 parts:

1. Typing Tutorial: The Typing Tutorial feature provides users with a comprehensive introduction to typing techniques. It gives interactive lessons that teach proper finger placement, hand positioning, and typing posture. Users will learn the basics of typing, including home row keys and finger assignment for each key.
2. Practice: In the Practice mode, users have the opportunity to improve their typing skills through a series of structured exercises. Users can select different types of typing exercises such as character type, word type, text type practice. The application tracks users' progress, including words per minute (WPM) and accuracy, to help them monitor their improvement over time.
3. Typing Game: The Typing Game feature adds challenging aspect to the learning experience. Users can test their typing skill in an exciting challenge. User will get a certain limited time and he have to play this typing game within this time. He will get a score based on his performance.

**2. Background Study**

A good typing skill is an important skill in this digital age. With an increasing reliance on computers and digital devices, efficient typing has become a fundamental skill in various professions. The project aims to provide a comprehensive solution for individuals seeking to improve their typing speed, accuracy, and technique. This background study explores the necessity and objectives of this Typing Tutor software project.

**2.1** Necessity of Typing Tutor Software:

1. Improving Digital Literacy: Typing fluency is essential for efficient work and effective communication in the modern digital world.
2. Productivity at Work: Faster data input, quicker document creation, and more effective communication are all made possible by enhanced typing skills in working environments.
3. Educational Benefits: Typing proficiency allows students to express their information properly without being constrained by slow typing speed, which is crucial for research projects, examinations, and assignments.

**2**

1. Skill Development: The Typing Tutor project helps users develop their skills by providing them with performance analysis, organized instructions, and a variety of practice tasks. This allows them to monitor their progress and make continuous improvements.
2. Personal and Professional Growth: Proficient typing skills are transferable across various domains, contributing to personal growth by boosting confidence and professional growth by enhancing employability.

**2.2** Objectives of Typing Tutor Software:

1. Improving Typing Speed: Enhancing users' typing speed through structured exercises and practice modules to enable faster typing.
2. Increasing Accuracy: Focusing on accuracy in typing to reduce errors and improve overall typing precision.
3. Teaching Proper Technique: Educating users on correct finger placement, posture, and typing techniques to maximize efficiency.
4. Customizable Learning Experience: Providing a personalized learning environment where users can choose specific exercises, and track their progress.
5. Multilingual Support: Including support for both Bangla and English language to make more useful to users.
6. Performance Analysis and Tracking: User can check his previous performance and can identify his weak zones.

# 3. Description of the Project

# 3.1 Login System:

# This login system allows user to access the application using their account info, also it helps to store the performance results of every user separately. As usual, there are two parts in this login system: Login and Sign Up. The Sign Up process allows users to create new accounts within the application. During Sign Up, users have to provide certain information such as a username and password. This information will be then stored securely in a text file. To store the password securely, I have used password encryption technique Caesar Cipher. The login process verifies the identity of users who already have registered accounts. The application then checks if the entered password matches the stored information in the text file. If the password is valid, the user gains access to their account.

# 3.2 Typing Tutorial:

# In this project, I have specifically designed the typing tutorial part to make it easy for beginners. In the English tutorial, I have divided all the keys into some different parts according to the fingers needed for typing, making the learning process easier and more interactive. Also in Bengali tutorial I have given Avro phonetic rules and corresponding instructions.

# 3.3 Typing Practice:

**3**

# In practice part, user can select different types of practice exercise like alphabet type, word type, text type. Also they can select different key areas. Users typing skill specially typing speed (WPM) and accuracy (in percentage) will be calculated here. Besides It will also show a histogram to represent the users’ weak areas where he makes the most mistakes. Users’ performance result will be saved in a text file so that he can check it later and a performance analysis can be done.

# 3.4 Game:

# The Typing Game will give a challenging environment to the learning experience. Users can test their typing speed and accuracy in a game mood. In this game user will get a specific time to play it. He will get a score based on his performance. Here I will use Edit Distance algorithm to give score. Users score will be then compared with the highest score which will be saved in a text file.

# 3.5 Performance history:

# Users’ typing exercises performance result will be stored in a text file. User can use the Performance history option to check the previous performance. Also it can show his improvement. User can understand which keys are his weak zones.

# 4. Implementation and Testing

# 4.1 Libraries and header files

# In this project I have to use various standard library or header files of C++. I created a root header file named “AllHeaderFile.h” and added all the libraries and header in this root header. Thus I can easily access these header from any cpp program within this project. Some of the most used headers and libraries are:

# <iostream>: Used for console input/output operations, reading from and writing to the standard input/output streams.

# <windows.h>: Used for Windows-specific functionalities like manipulating console screens (clearing screen, setting cursor position), creating windows, handling messages, working with files etc. functionalities available in the Windows operating system.

* <conio.h>: Used for console input operations to get characters from the keyboard, manipulate the console screen, etc. More specifically, conio.h header is used in this project to use the kbhit() and getche() function. kbhit() is used to check if a key has been pressed on the keyboard without blocking the program's execution. getche() is used to get a character from the console input without waiting for the Enter key to be pressed.

# <fstream>: Used for File handling operations like reading from a file, writing to a file, appending to a file, etc.

**4**

# <time.h>: Used for obtaining system time, measuring time intervals, formatting time and date, etc.

# <chrono>: This standard library is used for Precise time measurement, calculating time durations, timing operations, etc.

# <io.h> and <fcntl.h>: These headers are used for the \_setmode and \_fileno functions. \_setmode is employed to set the mode of a file stream, and \_fileno is used to obtain the file descriptor associated with a file stream. In this project, \_setmode is used to set the console to output UTF-16 text, ensuring that the console can correctly display wide characters. More specifically, these headers were necessary for working with Bangla text.

# <cstdlib>: Provides functions like malloc, free, exit, etc. used for memory management and general utilities.

# <bits/stdc++.h>: Used to simplify the process of including commonly used standard headers like <iostream>, <vector>, <algorithm>, <string>, etc., in a single include statement.

# 

# 4.2 Function Definition:

# In my project I have to create lots of function. To make them reusable, to avoid code repetition and to maintain these functions easily I have added all them in my root header file named “AllHeaderFile.h”. Thus I can easily use these function form any program within this project. Some of the most important functions are:

# void bangla\_homepage(): show the homepage for Bangla language.

# void english\_homepage(): show the homepage for English language.

# void drawKeyboard(): this function draw a keyboard on the console.

# void drawHistogram(char dataset[],int size,int frequency[]): this function draw a histogram of users’ error statistics on the console.

# void moveCursor(int x,int y): used to move the cursor to a specific position on the console.

# string RandomWord(): This function generates a random word.

# string bangla\_tounicode(wstring key): convert a bangla alphabet to it’s corresponding hexadecimal Unicode value.

# void loginsystem(): provides the login system interface.

# void history(): This function will show the users’ performance history.

# void keyDrill(char dataset[],int limit,int lessonNum,int partNum): Used to make a practice exercise on key typing.

# void wordDrill(char dataset[],int limit,int lessonNum,int partNum): Used to make a practice exercise on word typing.

# void bangla\_tutorial(): used to provide tutorial of Bangla typing.

# void bangla\_typing(): used to make practice exercise on Bangla typing.

# 

**5**

# 4.3 File I/O operation

# In this project, I have to use file handling operation for storing data to a text file. File I/O operations are used in these cases:

# - User info (username & password) save

# - user performance result save

# - typing exercise datasets store

# - typing tutorial instructions store

# In C++, ifstream and ofstream are classes in C++ used for file input and output operations. Using ifstream we can read data from a file. With ofstream, we can create a file if it doesn’t exist, or if it already exists, we can overwrite its contents or append new data to it. If we open a file in ofstream mode without specifying any additional flags, it will overwrite the existing content. To append data to the end of an existing file, we need to specify the std::ios::app flag when opening the file.

# *Figure 1: File write process*

# *Figure 2: File read process*

# 4.4 Bangla text I/O operation

In C++, the standard char can represent characters from the basic ASCII set, which covers English letters, digits, and some symbols. It's insufficient to represent the diverse set of characters in languages like Bangla, which have their own script and a larger set of characters.

In this case, we can use w\_char type to use wide character. Wide characters are capable of storing a wider range of characters because they use more memory compared to the standard char. The wchar\_t type use 16 or 32 bits per character whereas char type use 8 bits used by the char type. This increased memory usage with wchar\_t allows for the representation of a wider range of characters.

Functions like wprintf() and data types like std::wstring in C++ are used to handle wide characters and strings.

# In my project, there are some text file which contains Bangla text written in hexadecimal Unicode values. So to output them, first of all we have to set the console output mode to handle Unicode text in UTF-16 format. The hexadecimal code is 4 digit so we have to take 4 length substring. Then the hexadecimal value will be converted to integer value. Finally this integer value will be converted to wide character and it will be shown in the console.

**6**

# *Figure 3: Bangla Unicode text read form file and output*

# 4.5 Bangla text to Hex Unicode convert

# To check whether user give the input right, the Bangla character will be converted to it’s hexadecimal Unicode value. Then it will be compared with the uniocde value written in the text file. At first, the wide characters will be converted to its 4 digit Unicode hexadecimal representation. Then it will be converted to UTF-8 encoded form.

# *Figure 4: Bangla text to Unicode conversion*

# 4.6 Unicode hexadecimal value to Bangla text convert

# To show Bangla text in console we have to set the console to support printing Unicode characters in the Windows console. It enables the console to display 16-bit Unicode characters (wchar\_t) properly.

# *Figure 5: Hexadecimal Unicode value to Bangla text conversion*

**7**

# 4.7 Typing Time calculation

# Using the high-resolution clock method from the <chrono> library, this will measure the elapsed time between startTime and endTime. The duration between these two points is then stored in the duration variable, allowing you to analyze or output the elapsed time. Thus the time taken to complete a typing task is calculated.

# *Figure 6: Calculate typing time duration*

# 4.8 Typing Speed (WPM) Calculate

# There are two different ways to calculate speed here. Net and gross speeds. The amounts of key typed into words in a minute is known as gross speed. It is only a simple score that indicates how quickly you typed the keys. This truly illustrates the speed that may be achieved if a user makes no mistakes. The actual speed that provides the typing speed with errors calculated in the result is called net speed.

# *Figure 7: Calculate typing speed in WPM*

# 4.9 Accuracy Calculate

# The accuracy score indicates how many mistakes were made. The ratio of keys written properly to all keys typed is known as the accuracy percentage. Users have made less mistakes the higher the percentage. Error-free practice with 100% accuracy means no mistakes.

# Accuracy is calculated using standard word length. Accuracy percentage is always rounded down, for example 96.99 is considered 96%. Accuracy goals are defined as: 90% (Easy), 94% (Intermediary), 98% (Advanced).

**8**

# 4.10 Difficult keys analysis using histogram

# In a typing exercise, the error will be counted for corresponding keys. So a histogram can be made from the error frequency data.

# *Figure 8: Histogram drawing based on the error frequency*

# 4.11 Edit Distance Algorithm

# In the game part, a score will be given to make it more challenging. Here I defined the score giving process like that:

# *Figure 9: Score giving process in game part*

# If user type the correct word, his score will be increased by 5\*length of the word. But if his input his wrong, it will measure the edit distance between the given word and input word. Then user score will be decreased by 5\*edit distance. To find the edit distance I use the Dynamic Programming technique.

# 

**9**

# *Figure 10: Edit Distance DP Algorithm*

# 4.12 Random word generate:

# In typing practice, some randomly generated word will be given to user. Using the rand() function the random word will be generated.

# *Figure 11: Random word generate*

**10**

# 4.13 Cursor move

# Sometimes moving the cursor to a certain point in the console can be helpful to improve the look of the user interface.

# *Figure 12: Cursor moving in console screen*

# 4.14 Login System

# User have to give a username and password to create a new account. The login system will first verify if the username has already been used. If it’s available then user will give a password otherwise he have to give another unique username. These info will be saved in text file. To store the password securely it will be encrypted.

# *Figure 13: Caesar Cipher Algorithm implementation*

# In this project, I have used *Caesar* Cipher algorithm to encrypt-decrypt the password. The Caesar Cipher is a basic substitution algorithm that involves shifting every letter in the plaintext up or down the alphabet by a predetermined amount. For instance, 'A' would be encrypted as 'D', 'B' as 'E', and so forth with a shift of 3. The letters are moved by the predetermined amount during the encryption and decryption processes.

# *Figure 14: Sign Up process*

**11**

# The user must correctly enter their password and username in order to access their account. To verify the user's account, the input will be checked to the data which is stored in the text file.

# *Figure 15: Login process*

# 4.15 Performance history save

# Users’ performance info will be saved so that he can check it later. All the result will be stored in a text file.

# *Figure 16: Performance history save to text file*

# 5. User Interface

**12**

# 5.1 Login System:

# User have to select here to log in or sign up.

# Figure 17: Welcome page

# To create a new account, user have to give a unique username and a password. The username will be checked to see whether it is already in use. Username and password will be stored in text file, password will be stored in encrypted form.

# *Figure 18: Sign Up page*

# Providing correct username and password, user can access their account*.*

# *Figure 19: Login page*

# 5.2 Language Selection Page

# User have to select which language he wants to learn.

# *Figure 20: Language Selection page*

**13**

# 5.3 Home Page

# After selecting the language, user will get the corresponding homepage.

# . *Figure 21: Home page*

# 5.4 Tutorial Page

# Typing instructions are classified info some parts. User have to select which one he wants to learn. To get better learning experience it’s recommended to follow the given sequence.

# *Figure 22: Tutorial page (English)*

# *Figure 22: Tutorial page (Bangla)*

**14**

# 5.5 Typing Practice Page

# User have to select exercise type and set a time limit.

# *Figure 23: Practice type selection page*

# Some random word / key will be given. The user has the allotted time to finish this task. This is where his accuracy and speed of typing will be evaluated. Additionally, his mistakes will be monitored to identify the key in which he makes mistakes the most frequently.

# *Figure 24: Practice page (English)*

# *Figure 24: Practice page (Bangla)*

**15**

# *An analysis and performance result from the exercise will be provided. The errors made by users will be explained using a histogram.*

# *Figure 25: Practice result analysis*

# 5.6 Typing Game Page

# In this game user will get a certain time. Some words will be given and he has to type them upto the time. He will get a score. In this game part, he will just got a score which will be measured based on his performance, his speed and accuracy will be not calculated here.

# 

**16**

# *Figure 27: Fast Typer Game (Bangla)*

# 5.7 Statistics: Performance History

# *Figure 28: Performance history*

# 6. Challenges Faced

Developing a whole software project is never easy for someone who is doing it for the first time. A lot of challenges and obstacle I’ve faced during the project. Here are some significant challenges that I have faced:

1. Learning C++: During this project I had to learn C++ more deeply. I have to use some complex syntax which was previously unknow to me.
2. Handling a Large Project: In the beginning, I had no idea where to begin, how to design the project, or how to make connections between all the little components of my project. I had no experience on building such big projects. Then I divide my whole project into some modules. I was thinking about the process of connecting all the modules with a root module. Then I made a single C++ header files which defines all the modules. The header file can be accessed by any function that belonged to that project. Thus I have connect all the modules with each other easily.

**17**

1. File Handling Operations: In my project I have used text file for many cases. Users’ performance, users’ info like username and password is stored in text file. Also typing practice exercise are taken from text file. That’s why handling the files read write operation was a significant challenge for me.

1. Bangla Language Support: As C++ doesn't support Bangla language directly, it was not an easy task to add Bangla language support. But it supports Unicode. Then I try to use appropriate encoding standards between Unicode to Bangla character and thus I can effectively work with Bangla. Besides there are different keyboard layout for Bangla language. To keep Bangla typing easier to the beginner, here I used Avro Phonetic Keyboard.
2. Creating timer and time calculation: When user try typing exercises, he can set a specific time duration to practice. So I have to make a timer so that after that time the exercise is closed. That was initially looking hard task to me as I never worked with timer related task before. Then I tried to use C++ standard library <chrono> to work with time related operations.

**7. Conclusion**

In this project I tried to make a system which will help to improve typing skill especially to beginners. I tried to implement a well-designed interface on the console and give a good user experience. Throughout the project, I learnt a lot of things. Most importantly, I have learnt to manage a big project. I have also gained a deep understanding on C++ language, learnt some new syntax and operation of C++.

In future I can extend this project with more functionalities. Some of functionalities are described here:

* Enhanced User Interface: Improve the user interface design to make it more visually appealing and intuitive, providing a more immersive and enjoyable experience for users.
* Multiplayer Functionality: Implement multiplayer functionality to allow users to compete against each other in typing challenges, fostering a sense of competition and motivation.
* Cross-platform Compatibility: Adapt the game for compatibility with different platforms, including mobile devices, to reach a broader user base.
* Additional Game Modes: Introduce different game modes to provide a variety of typing challenges.

**Reference**

**18**

1. <https://www.unicode.org/charts/PDF/U0980.pdf>, Unicode Charts for Bangla character, 12/12/23
2. <https://www.rapidtables.com/convert/number/ascii-to-hex.html> , ASCII to HEX code converter online, 10/11/23
3. <https://www.coderstool.com/utf16-encoding-decoding> , UTF-16 Encoder-Decoder Online, 12/12/23
4. Bangla Typing with Avro Phonetic, PDF on Bangla typing using Avro phonetics by OmicronLab, 12/12/23
5. The Complete Reference C++ by Herbert Schild, 4th edition
6. <https://en.cppreference.com/w/cpp/chrono> , Time calculation operations, 12/12/23
7. <https://unicode.org/faq/utf_bom.html> , UTF variations, 15/12/23