GOVERNMENT POLYTECHNIC, AURANGABAD

(An Autonomous Institute of Government of Maharashtra)

"PERSUIT FOR EXCELLENCE"

PROJECT REPORT

ON

"LIMITLESS EDITOR"

SUBMITED BY

PRAKASH ZODGE (177062)

OM AGARWAL (177003)

YASH JAISWAL (177030)

VAISHNAVI NIGHVEKAR (177043)

GUIDED

BY

Prof. M. B.Dahival Sir

DEPARTMENT OF INFORMATION TECHNOLOGY

ACADEMIC YEAR 2019-20

DEPARTMENT OF INFORMATION TECHNOLOGY

GOVERNMENT POLYTECHNIC, AURANGABAD

(An Autonomous Institute of Government of Maharashtra)

CERTIFICATE

This is to certify that **Ms. Vaishnavi V. Nighvekar, MR. Yash Jaiswal, MR. Om Agarwal and Mr. Prakash Zodge has** successfully completed. There Project work titled **“LIMITLESS EDITOR”** during the academic year 2019-2020, in partial fulfillment of **Diploma in Information Technology** of Government Polytechnic, Aurangabad. Under the guidance of **Prof. M. B.Dahival sir .**To the best of my knowledge and belief this project work has not been submitted elsewhere.

Prof. M. B.Dahival Sir	**Prof. M. A. Dhaygude**
Project Guide	**H.O.D, I.T.**

Prof. F. A. KHAN

PRINCIPAL

ACKNOWLEDGEMENT

We sincerely thanks to our project guide **Prof. M. B. Dahival Sir** for giving time and helping us throughout our project. He has been a constant source of encouragement without which the project might not have been proceed. We are very grateful for his guidance.

We are expressing our profound gratitude to the Head of Department **Prof. M. A. Dhaygude Sir** for allowing us to proceed with our project and for giving us full freedom to access the lab facilities.

We also thankful to all the staff member of our department for their valuable suggestion in the process of this project work. Finally, yet importantly, we would like to express our thanks to our beloved parents for their blessing. Last but never the least let us thanks our friends and classmates for their help and cooperation for the successful completion of the project.

MR. PRAKASH ZODGE (147062)

MR. OM AGARWAL (177003)

MR. YASH JAISWAL (177030)

MS.VAISHNAVI NIGHVEKAR (177043)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

There are number of coding editors available in order to code in particular language. So every time language changes, mostly we need to download & install another editor or IDE for that language. This seems tedious process & storage wasting of system. So we proposed an editor named "Limitless Editor" which will offer multiple language support. The software specifically designed for Educational institutions where pc's seems to be low ended that is having low RAM and DISK SPACE.

User will be able to create new files & saved with particular extensions, and the program will run according to extension on single click. User will be able to import packages, methods & other basic syntax within the specific language just by clicking it from side panel.This project is essential to the people who are blind, as well as useful to people who are visually impaired as it provide text to speech facility in different voice.

CHAPTER 1
INTRODUCTION

1.1 WHAT IS IDE?

An **integrated development environment** (**IDE**) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger. Some IDEs, such as NetBeans and Eclipse, contain the necessary compiler, interpreter, or both; others, such as SharpDevelop and Lazarus, do not.

The boundary between an IDE and other parts of the broader software development environment is not well-defined; sometimes a version control system or various tools to simplify the construction of a graphical user interface (GUI) are integrated. Many modern IDEs also have a class browser, an object browser, and a class hierarchy diagram for use in object-oriented software development.

1.2 HOW IDE WORKS?

IDE’s are migration form of primitive text editors, which uses full functionality technologies that allow editing the code quickly and efficiently. An IDE typically contains a graphical user interface (GUI) to access the code editor, a compiler or interpreter and a debugger. An IDE starts with a model that translates into a suitable code.

An integrated development environment (IDE) makes working easy by providing facilities such as a source code editor, build automation tools and a debugger to programmers for software development. IDE makes easy to see a visual representation of the files and makes it more understandable for the user.

Depending on the kind of program, IDE can build and deploy the whole process easier. Using a good IDE makes compiling and debugging easier. You can compile and run the program, by clicking the Run button

1.3 LIMITLESS EDITOR

The **Limitless Editor** is a type of very simple editor specially designed for java, C++ and C is used in console programing and normal cpp and java programs. The screen for java editor is divided into three parts 1) the plain TextArea where you can easily write programs and can also adjust its size by expanding the window size, the default font size of this text area is 10 and default font style is calibre. 2) The second part of the screes contains the tree, where all the regular functions, class, package, and other keyword are defined. You can easily just double click on the tree to copy the corresponding name of the tree to the text area. 3) Suggestion Area or Autocomplete Area - The 3rd part of the screen contain the word completion or autocomplete, where application predicts the rest of word a user is typing. In this interfaces, users can typically click on the suggestion and accept one of several.

All this type of functions of the editor will help the user to type the program conveniently and efficiently. Rather than this functioning there is a status bar and a menubar in the editor window which will help you in other options.

Limitless Editor is a shareware cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

Limitless Editor is a full featured Text editor for editing local files or a code base. It includes various features for editing code base which helps developers to keep track of changes. Various features that are supported by Limitless are as follows −

- Auto Indentation
- File Type Recognition
- Sidebar containing tree facility
- Plug-in and Packages

Limitless editor is used as an Integrated Development Editor (IDE) like Visual Studio code and NetBeans. The Limitless Editor is compatible with various operating systems like Windows, Linux and MacOS.

Code editors allow the users to edit the code scripts and text documents with various shortcut keys.

It is often the first editor that newer programmers pick up because it works on all operating systems and it is far more approachable than Emacs, Vim or even PyCharm.

It is easy to get started in Limitless Editor because the menus and options are accessible by using a mouse. There are no different modes to learn like Vim's normal and insert modes. The keyboard shortcuts can be learned over time rather than all at once in the case of Vim or Emacs.

It works well for beginners as soon as they install it and then can be extended with many of the features provided by an IDE like PyCharm as a developer's skill level ramps up.

An additional bonus of using Limitless Editor as a Python developer is that plugins are written in Python. Python developers can extend this editor with their own programming language rather than learn a new language like Emacs' Elisp or Vim's Vimscript.

1.4 ADVANTAGES AND DISADVANTAGES OF LIMITLESS EDITOR

Advantages:-

- Provides “one click run” and “one click copy” functions.
- Provides easy way to run and type the program.
- Provides Format support with different font styles and sizes.
- Allow to run program in multiple languages
- Tree structure will help to differentiate every class, packages, and many functions.
- Provide inbuilt screen reader function which help blind person to code
- Etc.

Disadvantages:-

- Unable to perform the programs for servlet and jsp.
- Lack to different frameworks such as “struds” and “spring”
- Unable to identify the shortcuts for the save and open.
- It doesn’t support servlet & JSP.
- When you do JDBC, you need to set class path.
- Etc.

1.5 SCOPE OF LIMITLESS EDITOR

It gives a user-friendly framework for different types of programming languages, such as C, Java, and CPP. It contains a complete package including source code editor, build automation tools, debugger, compiler, interpreter and other features such as support for the version control system, auto-completion of keywords (where you start typing the name of a function or variable, it fills out the rest of the name), etc. It integrates project files, which you work on and includes version control of source files such as git repository.

It provides inclusive facilities to a programmer for the development of software.Below are some points, which describe why we should use this editor in the application development:

-It has the ability to debug your program and compile your code.

-It makes easy to see a visual representation of the location of program files.

-It provide support for external plug-ins, and you can use them by providing interfaces to external tools like debugging tools.

-It provide a console to see the execution result and if you find any errors, you can easily debug the errors and fix them. In C++ example, you can stop the program and check the value of variables.

-You can set breakpoints to pause the program when it reaches a certain line of code.

1.6 FEATURES OF LIMITLESS EDITOR

1. Powerful API & Package Eco-System:-

Limitless Editor has a powerful, Python API that allows plugins to augment built-in functionality. Package Control can be installed via the command palette, providing simple access to thousands of packages built by the community.

2. Customize Anything:-

Key bindings, menus, macros, completions and more - just about everything in it is customizable with simple JSON files. This system gives you flexibility as settings can be specified on a per-file type and per-project basis.

3. Split Editing:-

Get the most out of your wide screen monitor with split editing support. Edit files side by side, or edit two locations in the one file.

4. Instant Project Switch:-

Projects in Limitless Editor capture the full contents of the workspace, including modified and unsaved files. You can switch between projects in a manner similar to Goto Anything, and the switch is instant, with no save prompts - all your modifications will be restored next time the project is opened.

5. Performance:-

It is built from custom components, providing for unmatched responsiveness. From a powerful, custom cross-platform UI toolkit, it sets the bar for performance.

6. Cross Plat-Form:-

Limitless Editor is available for Mac, Windows and Linux. One license is all you need to use it on every computer you own, no matter what operating system it uses. It uses a custom UI toolkit, optimized for speed and beauty, while taking advantage of native functionality on each platform.

7. Keyboard Shortcuts:-

The amount of keyboard shortcuts in this editor are astounding. This is my other absolute best feature of Limitless Editor. The less I can move away from the home keys on my keyboard, the more efficient I can be.

8. Extensibility:-

Provides the ability to extend the platform to include additional features and functionalities.

9. Security:-

Ensures that the application itself is protected and assures data interacting with third-party plugins or integrations do not allow for unapproved access or privilege escalation.

CHAPTER 2
LITERATURE SURVEY

2.1 PURPOSE OF PROJECT

There are number of coding editors available in order to code in particular language. So every time language changes, mostly we need to download & install another editor or IDE for that language. This seems tedious process & storage wasting of system. So we proposed an editor named "Limitless Editor" which will offer multiple language support. The software specifically designed for Educational institutions where pc's seems to be low ended that is having low RAM and DISK SPACE.
User will be able to create new files & saved with particular extensions, and the program will run according to extension on single click. User will be able to import packages, methods & other basic syntax within the specific language just by clicking it from side panel.This project is essential to the people who are blind, as well as useful to people who are visually impaired as it provide text to speech facility in different voice.

2.2 SURVEY OF EXISTING EDITOR

2.2.1.Name of Programmer: - Parth Gaikwad

Q.1) Which editor do you like of any language?
Ans: None.

Q.2) Do you get any difficulty for typing your programs?

Ans: Yes, in notpad there are not much facilities.

Q.3) According to you which are the basics language that a person should learn?

Ans: C, C++, JAVA.

Q.4) Which language is most lengthy for typing?

Ans: JAVA

Q.5) Do you suggest any improvement in the editor that you are using?

Ans: Yes, it should show suggestion, error lines and many moreimprovements.

Q.6) Do you like to type the program in your editor?

Ans: Not much

Q.7) According to you what type of editor will a programmer will prefer?

Ans: That gives the most facilities to the user/programmer.

Q.8)	what type of difficulty you get in your editor?	
	1. Not much faster	2. Not iteractive
	3. Less facilities	4. None
Ans:	3. less facilities	

Q.9)	Will you try any another editor?	
	1. Yes	2. No
Ans:	1. Yes	

Q.10) Will you use LimitLess editor?

1. Yes	2. No

Ans: 1. Yes

2.2.2.Name of Programmer: **-**Rushikesh Oza

Q.1) which editor do you like of any language?

Ans: tdm gcc.

Q.2) Do you get any difficulty for typing your programs?

Ans: Yes, It's just like notepad with not much facilities.

Q.3) According to you which are the basics language that a person should learn?

Ans: C, C++.

Q.4) which language is most lengthy for typing?

Ans: C++.

Q.5) Do you suggest any improvement in the editor that you are using?

Ans: Yes, It should show suggestion, error lines.

Q.6) Do you like to type the program in your editor?

Ans: No.

Q.7) According to you what type of editor will a programmer will prefer?

Ans: That feels very easy to use.

Q.8) what type of difficulty you get in your editor?

1. Not much faster	2. Not iteractive
3. Less facilities	4. None

Ans: 1. not much faster

Q.9) Will you try any another editor?

1. Yes	2. No

Ans: 1. Yes

Q.10) Will you use LimitLess editor?

1. Yes	2. No

Ans: 1. Yes

2.2.3 Name of Programmer: - Shrikant Baheti

Q.1) which editor do you like of any language?

Ans: There are many editors but I like to use NET Beans for java andPycharm for python.

Q.2) Do you get any difficulty for typing your programs?

Ans: Yes, sometime I feel lengthy for typing programs.

Q.3) According to you which are the basics language that a person should learn?

Ans: C, C++, JAVA.

Q.4) which language is most lengthy for typing?

Ans: JAVA

Q.5) Do you suggest any improvement in the editor that you are using?

Ans: Yes, there can be many improvements such as drag and drop,auto speaking programs etc.

Q.6) Do you like to type the program in your editor?

Ans: Not Sure.

Q.7) According to you what type of editor will a programmer will prefer?

Ans: Any programmer will prefer a user friendly and easy to useEditor

Q.8) what type of difficulty you get in your editor?

1. Not much faster	2. Not iteractive
3. Less facilities	4. None

Ans: 1. Not much faster

Q.9) Will you try any another editor?

1. Yes	2. No

Ans: 1. Yes

Q.10) Will you use LimitLess editor?

1. Yes	2. No

Ans: 1. Yes

2.2.3. Name of Programmer: - Ganesh Bansa

Q.1) Which editor do you like of any language?

Ans: The Tourbo C Editor are very usefull

Q.2) Do you get any difficulty for typing your programs?

Ans: Yes

Q.3) According to you which are the basics language that a person should learn?

Ans: C, C++, JAVA.

Q.4) Which language is most lengthy for typing?

Ans: JAVA

Q.5) Do you suggest any improvement in the editor that you are using?

Ans: Yes,

1) There should be error line function in the editor.

2) GUImust me interactive.

Q.6) Do you like to type the program in your editor?

Ans: Sometimes

Q.7) According to you what type of editor will a programmer will prefer?

Ans: A user-friendly Editor will be very helpful for any programmer.

Q.8) What type of difficulty you get in your editor?

1. Not much faster	2. Not iteractive
3. Less facilities	4. None

Ans: 2. Not iterative

Q.9) Will you try any another editor?

1. Yes	2. No

Ans: 1. Yes

Q.10) Will you use LimitLess editor?

1. Yes	2. No

Ans: 1. Yes

2.3 ANALYSIS ON LITERATURE SURVEY

In this literature Survey we have collected data from many programmers in our locality and on the basis of that we have come up with this project. Because of more than 70% of programmers were in the favor of having an applications which will help the programmer to type the program easily and efficiently and must also have more new functions and facilities than the normal editor. In this survey we have found that it would be great help for the programmers if they get all this facilities easily and also free of cost. Hence we are planning on creating an application which would help the programmers to type there programs faster and in less efforts. Also importantly it will increase the use of **Limitless Editor** as the user will get more advance functions than any other editor. The need of Application in the programing world will be very helpful and can be used for the development of our nation towards technology. It will enhance the quality and belief of people on the editor so that they can get easy services form this editor.

The possibility of success of this Application is quiet guaranteed because the mark of 70% makes it clear for us. . Hence we are planning on creating an application which would help the programmers to reduce their work load as well as time. Also importantly it will increase the interest of the programmers as the **Limitless Editor** provide advance functions to their users.

CHAPTER 3
FEASIBILITY ANALYSIS

A Feasibility Study determines whether a project is worth doing. The process followed for making this determination is called a Feasibility Study. This type of study determines whether a project can and should proceed. Once it has been determined that a project is feasible, the analyst can proceed and prepare the project specifications that finalize the project specification.
The following are the various types of feasibility studies that can be undertaken.

3.1 TECHNICAL FEASIBILITY:-

This is concerned with specifying the equipments and the software to satisfy the user requirements. The technical needs of the system vary considerably but might include:

- The facility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of transactions at a specified speed.
- Facility to communicate data to a distant location.

Technical feasibility centers on the existing computer system, hardware, software etcetera and to what extent it can support the system. In examining the technical feasibility, the configuration of the system is given more importance than the actual hardware. The configuration should provide the complete picture of the system requirements, for example how many workstations are required and how these units are interconnected so that they would operate smoothly, etcetera. The result of the Technical Feasibility Study is the basis for the documents against which dealer and manufacturer can make bids. Specific hardware and software products can then be evaluated keeping in view the logical needs.

3.2 ECONOMIC FEASIBILITY:-

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design.

It is not done to analyze the new system. Using a Gantt chart schedule and part chart. We assumed that the benefit of the project is greater than the cost. So if we can develop the project easily then it is used for the evaluation of the proposed. We calculate the cost/benefit analysis and we assume that the benefit is feasible so we start developing the project. It is an analysis of the cost to be incurred in the system and benefits the derivable from the system. An economic Feasibility Study should demonstrate the net benefit of the proposed course of action in the context of direct and indirect benefits and costs to the organization and to the public as a whole. It should be required for both pilot and long-term activities, plans and projects.

3.3 OPERATIONAL FEASIBILITY:-

It determines how acceptable the software is within the organization. The evaluations must then determine the general attitude and skills. Such restriction of the job will be acceptable. To the users are enough to run the proposed budget, hence the system is supposed to the feasible regarding all except of feasibility. In operational feasibility, we attempt to enusre that every user can access the system easily. We develop a menu that users can easily access and we provide shortcut keys.

We show a proper error message when any mistakes are made in the program. We provide help and a guideline menu to help the user.

Changes in the ways individuals are organized into groups may then be necessary and the groups may now compete for economic resources with the needs of stabilized ones by converting a number in a file in software.

3.4 BEHAVIORAL FEASIBILITY:-

Normal human psychology of human beings indicate that people are resistant to change and computers are known to facilitate change. Any project formulations should consider this factor also. Before the development of the Project titled "Delhi Metro", the need to study the feasibility of the successful execution of the project was felt and thus the following factors are considered for a Feasibility Study.

- Need Analysis.
- Provide the users information pertaining to the preceding requirement.
- Feasibility Study Report

The result of the Feasibility Study provides us with the following facts:

• The automated system would increase the efficiency of the system.

• The automated system would increase customer's satisfaction.

• The automated system has many requirements such as Efficiency cost effectiveness, prompt service, Reliability.

• The automated system would add to the security features of the system

• The automated system should be simple to use, incorporate all necessary services and maintainable.

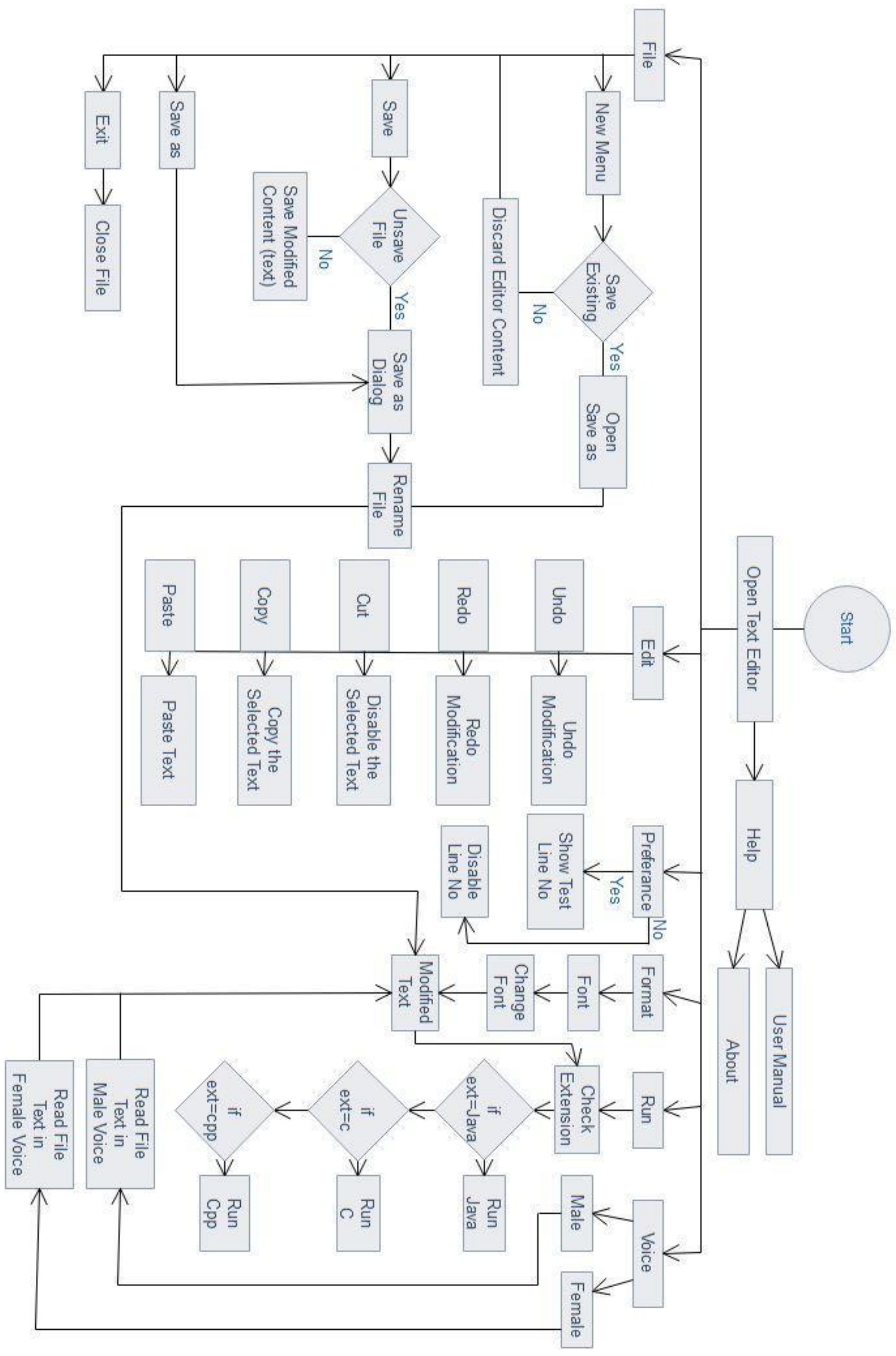
• This will cause some changes in the organization. These are:

Change in staffing policies: present employees will need to be sent for training.

New employees to be recruited will need to have the knowledge of the automated system.

Chapter 4
Design of Project

Fig 4.1 Design of project



CHAPTER 5
USE CASE DIAGRAM OF PROJECT

5.1 WHEN TO APPLY USE CASE DIAGRAMS

A use case diagram doesn't go into a lot of detail—for example, don't expect it to model the order in which steps are performed. Instead, a proper use case diagram depicts a high-level overview of the relationship between use cases, actors, and systems. Experts recommend that use case diagrams be used to supplement a more descriptive textual use case.

UML is the modeling toolkit that you can use to build your diagrams. Use cases are represented with a labeled oval shape. Stick figures represent actors in the process, and the actor's participation in the system is modeled with a line between the actor and use case. To depict the system boundary, draw a box around the use case itself.

UML use case diagrams are ideal for:

- Representing the goals of system-user interactions
- Defining and organizing functional requirements in a system
- Specifying the context and requirements of a system
- Modeling the basic flow of events in a use case

5.2 USE CASE DIAGRAM COMPONENTS

To answer the question, "What is a use case diagram?" you need to first understand its building blocks. Common components include:

- **Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your Editor. They must be external objects that produce or consume data.

- **System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.
- **Goals:** The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

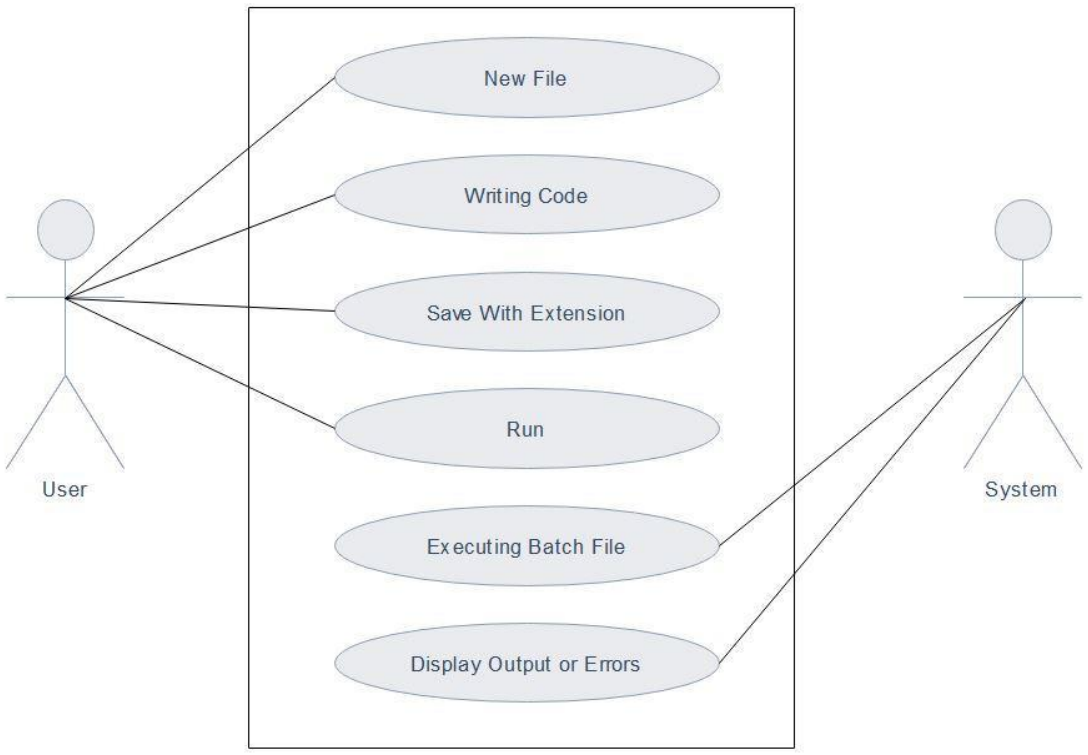


Fig 5.2 Use Case Diagram

CHAPTER 6
USER MANUAL

The Limitless Editor is a type of very simple editor specially designed for java, C++ and C is used in console programing and normal C, C++ and java programs. The screen for java editor is divided into three parts

1) Plain Text Area –

The plain Text Area where you can easily write programs and can also adjust its size by expanding the window size, the default font size of this text area is 10 and default font style is calibre.

2) Tree Area –

The second part of the screes contains the tree, where all the regular functions, class, package, and other keyword are defined. You can easily just double click on the tree to copy the corresponding name of the tree to the text area.

3) Suggestion Area or Autocomplete Area -

The 3rd part of the screen contain the word completion or autocomplete, where application predicts the rest of word a user is typing. In this interfaces, users can typically click on the suggestion and accept one of several.

All this type of functions of the editor will help the user to type the program conveniently and efficiently. Rather than this functioning there is a Tool bar and a menu bar in the editor window which will help you in other options.

- Toolbar –

A toolbar is a set of icons or buttons that are part of a software program's interface or an open window. When it is part of a program's interface, the toolbar typically sits directly under the menu bar. Our editor includes a toolbar that allows you to cut, copy, paste, undo, redo and create a new file.

- Menu Bar-

This is a most important element of the editor window as it contains every functioning of the editor. You can move along the menu bar moving the cursor and pressing the right key. Alternatively you can use shortcut keys described after menu name below.

When you have selected your menu choice, you can select the option required by use of the UP and DOWN cursor keys and when Thunder announced your choice , press the Enter key and your selection will be activated. Let's see more detailed description of the Menu Bar Items.

6.1 FILE MENU-

This is where you can "Open, Close, Save, Rename and Create New "files. An example of such list is shown below and a very brief description of its function.-

Name	**Key**	**Description**
New	Ctrl+N	To initialize a new document
Open	Ctrl + O	Allows you to open an existing document
Save	Ctrl + S	Allows you to save a named document without clearing the document. If you are working on a document for a long time it is recommended that you save you work at regular intervals so as to minimise the loss of data.
Save As	Ctrl + Shift +S	This gives you the option of saving your document under a new name or different format.

Exit	Ctrl + X	You can leave the Editor with or without saving your document, be careful. Have some fun trying out these options. Please be careful when experimenting, we do not want you to lose any vital data.

Table 6.1.1 : File Menu

6.2 EDIT :-

This is where you can "Cut, Copy, and Paste" characters, words in your documents.An example of such a list is shown below and a very brief description of its function. The shortcut key combinations are shown in Key column following the Name option.

Name	**Method to select**	**Description**
Cut	• Usekeyboard shortcut **Ctrl-X**. • Choose Cut from the Edit menu. • Click on Cut button in the toolbar. • Right-click on the selection and choose Cut from the context menu.	The selection will be Cut to the clipboard buffer. To remind you of this, the border of the selection will be shown as "marching ants". The selection will be moved to a new location as soon as you choose Paste command as described below.
Copy	• Use keyboard shortcut **Ctrl-C**. • Choose Copy from the Edit menu.	The selection will be copied to the clipboard buffer. The selection will be copied to a new location as soon as you

	• Click on Copy button in the toolbar. • Right-click on the selection and choose Copy from the context menu.	choose Paste command as described below.
Paste	• Use keyboard shortcut **Ctrl-V**. • Choose Paste from the Edit menu. • Click Paste button in the toolbar. • Right-click on the cell and choose Paste from the context menu.	If you are pasting a selection which was cut to the clipboard buffer, all cell references in all formulas will remain unchanged. The original selection will be removed from the workbook and the clipboard buffer will be cleared.
Undo	• Use keyboard shortcut **Ctrl+ Z**. • Choose Undo from the Edit menu. • Click Undo button in the toolbar. Right-click on the cell and choose Undo from the context menu.	To reverse your last action, press CTRL+Z. You can reverse more than one action.
Redo	• Use keyboard shortcut **Ctrl+ R**. • Choose Redo from the Edit menu. • Click Redo button in the toolbar.	To reverse your last Undo, press CTRL+R. You can reverse more than one action that has been undone. You can use Redo command only after Undo

	• Right-click on the cell and choose Redo from the context menu.	command.

Table 6.2.1 : Edit

6.3 PREFERENCES –

The preference menu consists of only one Menu Item i.e. Toggle Line Number generally use to hide and show the line number.

Name	Method to select	Description
Toggle Line Number	Choose Toggle Line Number from the Preferences menu	Use to display or hide line numbers on left hand side of the Plain Text Area.

Table 6.3.1 : Preferences

6.4 FORMAT MENU-

The format command is commonly used among the other commands on the Limitless editor menu bar. It helps us to format documents in various ways to achieve the desired effects. The following options are available on the format menu.

Name	Method to select	Description
Font	• Use keyboard shortcut **Ctrl+ F**. • Choose Font from the Format menu.	This command enables us to format the Font by changing the font, font style, size, font colour, underline style, underline colour and you can also add some special effects like strikethrough and Underline.

Table 6.4.1 : Format Menu

6.5 RUN MENU-

This is the important menu of our editor which have two sub menu, they are Run java program and Run C/C++ program. As there name suggest what this options actually do is to run the java program C program and the C++ program separately.

Name	Method to select	Description
Run Java Program	• Use keyboard shortcut **Shift+ J**. • Choose Run Java program from the Run menu.	As there name suggest what this options actually do it will run currently save java program if the extension of the saved program is .java else it will run the last saved file .
Run C Program	• Use keyboard shortcut **Shift+ C**. • Choose Run C program from the Run menu.	It will run currently save C program if the extension of the saved program is .c and run C++ program if the extension of the saved program is .cpp else it will run the last save file of C/C++

Table 6.5.1 : Run Menu

6.6 VOICE MENU-

This menu will convert the text into speech and read it aloud in computerize voice and helps the blind people to read the code. This menu contains two more sub menu they are as follows-

Name	Method to select	Description
Male	• Use keyboard shortcut **Ctrl +Shift + M**. • Choose Malefrom the Voice menu.	It will convert the text into speech and read the converted text in Male voice

Female	• Use keyboard shortcut **Ctrl +Shift + F**. • Choose Femalefrom the Voice menu.	It will convert the text into speech and read the converted text in Female voice

Table 6.6.1 : Voice Menu

CHAPTER 7
COSTING

We are not providing any pricing information for this product or service. This is common practice for software vendors and service providers. The pricing insights provided here are based on user reviews and are intended to give you an indication of value. Alternatively, contact us to obtain current pricing

CHAPTER 8
SOURCE CODE OF PROJECT

8.1 CODE OF EDITOR:-

```
import wx
import wx.adv
import os
import wx.stc as stc
import pyautogui
import tree
import speech_recognition as sr
import pyttsx3

from gtts import gTTS

global name
name = 0
global fileloc
fileloc = 0




screen_width, screen_height = pyautogui.size()
ls = []
ls1 = []
global flag
flag = False
lista = ['public static void main{}', 'actions', 'static', 'void', 'main',
'protected',"#include<stdio.h>", "#include<conio.h>","if(condition){}",

"#include<string.h>","#include<stdlib.h>","#include<math.h>","#include<time.h>","#
include<ctype.h>","#include<stdarg.h>",
"#include<signal.h>","#include<setjmp.h>","#include<locale.h>","#include<errno.h>
",'assert',"#include<assert.h>","printf("")",
"scanf("")","getchar()","putchar()","fopen()","fclose()","if(condition){}",'while(conditi
on){}"for(condition){}','do()',

'do{}while(condition);',"printf("")","scanf("")","getchar()","putchar()","putchar()","fop
en()","fopen()","fclose()","fclose()","clrscr()",

"getch()","getch()",'while(Condition){Statement(s)Increment/Decrement;}','for(initiali
zation; condition; incOrDec){ }',
        'import java.io.*;','import java.util.*;','import java.awt.*;','import
java.net.*;','import java.awt.event;','import java.lang.*;',
        'import java.applet.*;','import javax.swing.*;','import
javax.swing.event;,Label()','Label l = new Label("");','return 0'
        ,'Button b = new Button("");\n','Choice c = new Choice();','TextField t = new
TextField("");','TextArea();','Checkbox obj = new Checkbox();\n',
        'Checkbox obj = new Checkbox(String label, boolean state, CheckboxGroup
obj);','cin>>my_value;','cout<<statement;','List s = new List();\n',
        'MenuBar mb = new MenuBar();\n','Menu m = new Menu("");\n','MenuItem mi =
new MenuItem("");\n','JLabel obj = new JLabel("");\n',
        'JButton b = new JButton("");\n','JTextField t = new JTextField("");\n','JTextArea
t = new JTextArea("");\n',
        'JComboBox C = new JComboBox();\n','JCheckBox c = new JCheckBox("1");\n'
        ,'JTree jt=new JTree(DefaultMutableTreeNodeObjectName);\n','JPasswordField p
= new JPasswordField();\n'
        'JMenuBar mb = new JMenuBar();\n','JMenu m = new JMenu("");\n','JMenuItem
mi = new JMenuItem("");\n','int main(){}',"__"
          ]


class Editor(wx.Frame):
    lineNumbersEnabled = True

    def __init__(self, parent, title):
        self.leftMarginWidth = 25
        wx.Frame.__init__(self, parent, title=title, size=(screen_width, screen_height))

        self.CreateStatusBar()
        self.StatusBar.SetBackgroundColour((220, 220, 220))

        panel1 = wx.Panel(self)
        self.control = stc.StyledTextCtrl(panel1, style=wx.TE_MULTILINE | wx.TE_WORDWRAP)
        self.control1 = stc.StyledTextCtrl(panel1, style=wx.TE_MULTILINE | wx.TE_WORDWRAP)

        self.control.CmdKeyAssign(ord("+"), stc.STC_SCMOD_CTRL, stc.STC_CMD_ZOOMIN)
        self.control.CmdKeyAssign(ord("-"), stc.STC_SCMOD_CTRL, stc.STC_CMD_ZOOMOUT)

        self.control.SetViewWhiteSpace(False)
        self.control.SetMargins(5, 0)
        self.control.SetMarginType(1, stc.STC_MARGIN_NUMBER)
        self.control.SetMarginWidth(1, self.leftMarginWidth)

        self.control.Bind(stc.EVT_STC_CHANGE, self.sug)
        self.control.Bind(wx.EVT_KEY_UP, self.dell)
        self.control1.Bind(wx.EVT_LEFT_DCLICK, self.copy)
        self.control1.SetReadOnly(True)
        self.tree1 = tree.Tree(panel1, wx.TR_HIDE_ROOT)
        box = wx.BoxSizer(wx.HORIZONTAL)
        box.Add(self.tree1, 1, wx.EXPAND, 20)
        box.Add(self.control, 4, wx.EXPAND, 20)
        box.Add(self.control1, 2, wx.EXPAND, 20)
        panel1.SetSizer(box)
        panel1.Fit()
        filemenu = wx.Menu()

menuNew=wx.MenuItem(filemenu,wx.ID_NEW,'&New\tCtrl+N',kind=wx.ITEM_N
ORMAL)
        menuNew.SetBitmap(wx.Bitmap('Resources/newfile.png'))
        filemenu.Append(menuNew)

        menuOpen =wx.MenuItem(filemenu,wx.ID_OPEN, '&Open\tCtrl+O', "Open an
Existing Document")
        menuOpen.SetBitmap(wx.Bitmap('Resources/openfile.png'))
        filemenu.Append(menuOpen)

        menuSave = wx.MenuItem(filemenu,wx.ID_SAVE, '&Save\tCtrl+S', "Save the
Current Document")
        menuSave.SetBitmap(wx.Bitmap('Resources/save.png'))
        filemenu.Append(menuSave)

        menuSaveAs = wx.MenuItem(filemenu,wx.ID_SAVEAS,'&Save
As\tCtrl+Shift+S', "Save the New Document")
        menuSaveAs.SetBitmap(wx.Bitmap('Resources/saveas.png'))
        filemenu.Append(menuSaveAs)

        filemenu.AppendSeparator()
        menuClose = wx.MenuItem(filemenu,wx.ID_EXIT, '&Exit\tCtrl+Alt+X', "Close
the Application")
        menuClose.SetBitmap(wx.Bitmap('Resources/exit.png'))
        filemenu.Append(menuClose)

        editmenu = wx.Menu()
        menuUndo = wx.MenuItem(editmenu,wx.ID_UNDO, "&Undo\tCtrl+Z", "Undo
Last Action")
        menuUndo.SetBitmap(wx.Bitmap('Resources/undo.png'))
        editmenu.Append(menuUndo)
        menuRedo = wx.MenuItem(editmenu,wx.ID_REDO, "&Redo\tCtrl+R", "Redo
Last Action")
        menuRedo.SetBitmap(wx.Bitmap('Resources/redo.png'))
        editmenu.Append(menuRedo)
        editmenu.AppendSeparator()
        menuCut = wx.MenuItem(editmenu,wx.ID_CUT, '&Cut\tCtrl+X', "Cut Selected
Text")
        menuCut.SetBitmap(wx.Bitmap('Resources/cut.png'))
        editmenu.Append(menuCut)
        menuCopy = wx.MenuItem(editmenu,wx.ID_COPY, "&Copy\tCtrl+C", "Copy
Selected Text")
        menuCopy.SetBitmap(wx.Bitmap('Resources/copy.png'))
        editmenu.Append(menuCopy)
        menuPaste = wx.MenuItem(editmenu,wx.ID_PASTE, "&Paste\tCtrl+V", "Paste
the Text from Clipboard")
        menuPaste.SetBitmap(wx.Bitmap('Resources/paste.png'))
        editmenu.Append(menuPaste)
        formatmenu = wx.Menu()
        menuFont = wx.MenuItem(formatmenu,wx.ID_ANY, "&Font\tCtrl+F", "Change
the Font Settings")
        menuFont.SetBitmap(wx.Bitmap('Resources/font.png'))
        formatmenu.Append(menuFont)

        runmenu = wx.Menu()
        menuRunJava = wx.MenuItem(runmenu,wx.ID_ANY, "&Run Java
Program\tShift+J", "Run Java Program")
        menuRunJava.SetBitmap(wx.Bitmap('Resources/jar.png'))
        runmenu.Append(menuRunJava)
        menuRunC = wx.MenuItem(runmenu,wx.ID_ANY, "&Run C Program\tShift+C",
"Run C Program")
        menuRunC.SetBitmap(wx.Bitmap('Resources/class.png'))
        runmenu.Append(menuRunC)

        prefmenu = wx.Menu()
        menulinenumbers = wx.MenuItem(prefmenu,wx.ID_ANY, "&Toggle Line
Numbers", "Show/Hide Line Numbers Column")
        menulinenumbers.SetBitmap(wx.Bitmap('Resources/options.png'))
        prefmenu.Append(menulinenumbers)

        helpmenu = wx.Menu()
        menuhowto = wx.MenuItem(helpmenu,wx.ID_ANY, "&How To..\tCtrl+H", "Get
Help Using the Editor")
        menuhowto.SetBitmap(wx.Bitmap('Resources/how.png'))
        helpmenu.Append(menuhowto)
        menuabout   =   wx.MenuItem(helpmenu,wx.ID_ABOUT,   "&About\tShift+A",
"Read About Editor and its Making")
        menuabout.SetBitmap(wx.Bitmap('Resources/about.png'))
        helpmenu.Append(menuabout)

        voicemenu = wx.Menu()
        menuvaish  =  wx.MenuItem(voicemenu,wx.ID_ANY,  "&Female\tCtrl+Shift+F",
"Voice")
        menuvaish.SetBitmap(wx.Bitmap('Resources/female.png'))
        voicemenu.Append(menuvaish)
        menuvaish2  =  wx.MenuItem(voicemenu,wx.ID_ANY,  "&Male\tCtrl+Shift+M",
"voice")
        menuvaish2.SetBitmap(wx.Bitmap('Resources/male.png'))
        voicemenu.Append(menuvaish2)

        menubar = wx.MenuBar()
        menubar.Append(filemenu, '&File')
        menubar.Append(editmenu, '&Edit')
        menubar.Append(prefmenu, '&Preferences')
        menubar.Append(formatmenu, '&Format')
        menubar.Append(runmenu, '&Run')
        menubar.Append(voicemenu, '&Voice')
        menubar.Append(helpmenu, '&Help')
        self.SetMenuBar(menubar)
        self.Bind(wx.EVT_MENU, self.newproj, menuNew)
        self.Bind(wx.EVT_MENU, self.open1, menuOpen)
        self.Bind(wx.EVT_MENU, self.save_as, menuSaveAs)
        self.Bind(wx.EVT_MENU, self.save, menuSave)
        self.Bind(wx.EVT_MENU, self.OnUndo, menuUndo)
        self.Bind(wx.EVT_MENU, self.OnRedo, menuRedo)
        self.Bind(wx.EVT_MENU, self.OnCut, menuCut)
        self.Bind(wx.EVT_MENU, self.OnCopy, menuCopy)
        self.Bind(wx.EVT_MENU, self.OnPaste, menuPaste)
        self.Bind(wx.EVT_MENU, self.run, menuRunJava)
        self.Bind(wx.EVT_MENU, self.runc, menuRunC)
        self.Bind(wx.EVT_MENU, self.howto,menuhowto)
        self.Bind(wx.EVT_MENU, self.OnToggleLineNumbers, menulinenumbers)
        self.Bind(wx.EVT_MENU, self.Fontbox, menuFont)
        self.Bind(wx.EVT_TREE_ITEM_ACTIVATED, self.select)
        self.Bind(wx.EVT_MENU, self.tts, menuvaish)
        self.Bind(wx.EVT_MENU, self.SpeechDef, menuvaish2)
        self.control.Bind(wx.EVT_KEY_UP, self.UpdateLineCol)
        self.control.Bind(wx.EVT_LEFT_UP, self.OnLeftUp)
        self.Bind(wx.EVT_MENU,self.OnAbout, menuabout)
        toolbar1 = wx.ToolBar(self, -1)
        self.ToolBar = toolbar1
        t1 = toolbar1.AddTool(wx.ID_ANY, '', wx.Bitmap('Resources/newfile.png'))
        t2 = toolbar1.AddTool(wx.ID_ANY, '', wx.Bitmap('Resources/cut.png'))
        t3 = toolbar1.AddTool(wx.ID_ANY, '', wx.Bitmap('Resources/copy.png'))
        t4 = toolbar1.AddTool(wx.ID_ANY, '', wx.Bitmap('Resources/paste.png'))
        t5 = toolbar1.AddTool(wx.ID_ANY, '', wx.Bitmap('Resources/undo.png'))
        t6 = toolbar1.AddTool(wx.ID_ANY, '', wx.Bitmap('Resources/redo.png'))
        toolbar1.Realize()
        # End of Toolbar
        self.Bind(wx.EVT_MENU, self.newproj, t1)
        self.Bind(wx.EVT_MENU, self.OnCut, t2)
        self.Bind(wx.EVT_MENU, self.OnCopy, t3)
        self.Bind(wx.EVT_MENU, self.OnPaste, t4)
```

```
        self.Bind(wx.EVT_MENU, self.OnUndo, t5)
        self.Bind(wx.EVT_MENU, self.OnRedo, t6)
        self.Show()

    def sug(event,e):
        event.control1.SetReadOnly(False)
        global flag
        global text
        aa = float(event.control.GetCurrentPos())
        if aa>0.0:
            a = str(event.control.GetRange(float(int(aa - 1.0)),float(event.control.LastPosition)))

        else:
            a = str(event.control.GetRange(float(int(aa - 0.1)), float(event.control.LastPosition)))

        event.control1.DeleteRange(0.0, float(event.control1.LastPosition))
        if a.__contains__(" "):
            if aa > 0.0:
                a = str(event.control.GetRange(float(int(aa - 1.0)), float(event.control.LastPosition)))
            else:
                a = str(event.control.GetRange(float(int(aa - 0.1)), float(event.control.LastPosition)))
            i = 0
            b1 = ""
            b2 = ""
            ls = a.split(" ")
            a1 = ls[ls.__len__() - 1]
            a1 = a1.rstrip()
            ls1 = a1.split("\n")
            a2 = ls1[0]
            text = a2
            event.control1.DeleteRange(0.0, float(event.control1.LastPosition))
            while (i < lista.__len__() - 1):
                b = str(lista[i])

                if b.__len__() > 1 and a2.__len__() == 2:
                    b1 = b[0] + b[1]
                if b.__len__() > 2 and a2.__len__() == 3:
                    b2 = b[0] + b[1] + b[2]

                if a2.__contains__(b[0]) and a2.__len__() == 1:
                    event.control1.InsertText(0.0, lista[i] + "\n")
                if a2.__contains__(b1) and a2.__len__() == 2:
                    event.control1.InsertText(0.0, lista[i] + "\n")
                if a2.__contains__(b2) and a2.__len__() == 3:
                    event.control1.InsertText(0.0, lista[i] + "\n")

                i = i + 1
            a = ""

        else:
            b1 = ""
            b2 = ""
            z=str(event.control.GetRange(0.0, float(event.control.LastPosition)))
            event.control1.DeleteRange(0.0, float(event.control1.LastPosition))
            if flag == True:
                if z.__len__()==2:
                    a = str(event.control.GetRange(aa - 1.0, float(event.control.LastPosition)))
                elif z.__len__()==3:
                    a = str(event.control.GetRange(aa - 2.0, float(event.control.LastPosition)))
                else:
                    a = str(event.control.GetRange(aa - 0.1, float(event.control.LastPosition)))
                flag = False
            else:
                if z.__len__() == 2:
```

```
        a = str(event.control.GetRange(aa - 1.0, float(event.control.LastPosition)))
    elif z.__len__() == 3:
        a = str(event.control.GetRange(aa - 2.0, float(event.control.LastPosition)))
    else:
        a = str(event.control.GetRange(aa - 0.1, float(event.control.LastPosition)))

i = 0
text = a
while (i < lista.__len__() - 1):
    b = str(lista[i])
    if b.__len__() > 1 and a.__len__() == 2:
        b1 = b[0] + b[1]
    if b.__len__() > 2 and a.__len__() == 3:
        b2 = b[0] + b[1] + b[2]



    if a.__contains__(b[0]) and a.__len__() == 1:
       # print(1)
        event.control1.InsertText(0.0, lista[i] + "\n")

    if a.__contains__(b1) and a.__len__()  == 2:
       # print(2)
        event.control1.InsertText(0.0, lista[i] + "\n")

    if a.__contains__(b2) and a.__len__()  == 3:
       # print(3)
        event.control1.InsertText(0.0, lista[i] + "\n")

    i = i + 1

if a.__contains__(" ") or a.__len__() == 1:
    event.control1.DeleteRange(0.0, float(event.control1.LastPosition))
a = ""
event.control1.SetReadOnly(True)
```

```
    def dell(event,e):
        if e.GetKeyCode()==13:
            print("delete wala")
            event.control1.SetReadOnly(False)
            event.control1.DeleteRange(0.0, float(event.control1.LastPosition))
            global flag
            flag = True
            event.control1.SetReadOnly(True)
            e.Skip()
        else:
            pass

    def copy(event,e):
        global text
        print("copy wala")
        event.control1.SetReadOnly(False)
        i = 0
        ind = float(event.control.GetCurrentPos())
        a          =          event.control1.GetRange(float(event.control1.GetCurrentPos()),
float(event.control1.LastPosition))
        ls = a.split("\n")
        b = str(event.control1.GetRange(0.0, float(event.control1.LastPosition)))
        ls1 = b.split("\n")
        while (True):
            if ls[1] == ls1[i]:
                break
            i = i + 1
        c = ls1[i - 1]
        c1 = ""
        ind2 = 0.0
        ii = str(ind)
        if              str(event.control.GetRange(float(int(ind              -              1.0)),
float(event.control.LastPosition))).__contains__(" "):
```

```
            if text.__len__() == 1:
                c1 = c[0]
                ind2 = ind - 1.0
            if text.__len__() == 2:
                c1 = c[0] + c[1]
                ind2 = ind - 2.0
            if text.__len__() == 3:
                c1 = c[0] + c[1] + c[2]
                ind2 = ind - 3.0




        else:
            if text.__len__() - 1 == 1:
                c1 = c[0]
                ind2 = ind - 1.0
            if text.__len__() - 1 == 2:
                c1 = c[0] + c[1]
                ind2 = ind - 2.0
            if text.__len__() - 1 == 3:
                c1 = c[0] + c[1] + c[2]
                ind2 = ind - 3.0

        if text.__contains__(c1):
            event.control.Replace(ind2-1, ind, c)
        event.control1.SetReadOnly(True)




    def save(self, e):
        global name
        global fileloc
        print(type(name))
        print(type(fileloc))
        if (name == 0):
            if (fileloc == 0):
                initial_dir = r"C:\Users\vaish"
                # the filetype mask (default is all files)
                mask = [("Text and Python files", "*.txt *.py "),
                        ("C files", '*.c'),
                        ("cpp files", '*.cpp'),
                        ("HTML files", "*.htm"),
                        ("All files", "*.*")]
                dig = wx.FileDialog(self, "Save File As", initial_dir, "Untitled", "*.*",
                                    wx.FD_SAVE | wx.FD_OVERWRITE_PROMPT)
                if (dig.ShowModal() == wx.ID_OK):
                    name = dig.GetFilename()
                    dirname = dig.GetDirectory()
                    text = str(self.control.GetValue())

                    # location of file
                    fileloc = dig.GetPath()
                    print("fileloc", fileloc)
                    file = open(os.path.join(dirname, name), "w")
                    file.write(text)
                    # path of file
                    f = name
                    print('f', f)
                    file.close()
                    # yaha milta hai.txt
                    e = os.path.splitext(f)
                    ext = e[1]
                    print(ext)
                    ch = ext
                    # abhi ke leye aapne jaha bat file save ki vaha ka path dalo uska solution
    hai //fileloc se hota vo mai kal
                    # send karti
                    if (ext == '.java'):
```

```
ob = open("Resources\\file.bat", "r")
dat = ob.read()
ob.close()
ob = open("Resources\\file.bat", "r")
data = ob.readlines()
ob.close()
ob = open("Resources\\file.bat", "a")
n = f + "\n"

print('data:', data)
print('dat:', dat)
for j in data:
    # bat file mai jo data leya hai vo space se split keya hai so jaha space
hai vo list mai jaye
    ls = list(j.split(" "))
    if ls[0] == "javac":
        print(ls[0])
        # so ls of 1 pe sib file ka nam or extension aayega
        a = ls[1]
        dat = dat.replace(a, str(f) + "\n")
        print('dat1' + dat)
    if ls[0] == "java":
        ll = f.split(".")
        a = ls[1]

        dat = dat.replace(a, ll[0] + "\n")
        print('dat2' + dat)
    if ls[0] == "cd":
        a = ls[1]
        print('a' + a)
        s = dat[0] + dat[1]
        print('s' + s)
        s1 = fileloc[0] + fileloc[1]
        print('s1' + s1)
```

```
                dat = dat.replace(a, str(fileloc) + "\n")
                dat = dat.replace(f, "", 1)
                dat = dat.replace(s, s1, 1)
                print('dat3' + dat)

        ob.close()
        print(dat)
        ob = open("Resources\\file.bat", "w")
        ob.write(dat)
        ob.close()

    if (ext == '.c' or ext == '.cpp'):
        ob = open("Resources\\c.bat", "r")
        dat = ob.read()
        ob.close()
        ob = open("Resources\\c.bat", "r")
        data = ob.readlines()
        ob.close()
        ob = open("Resources\\c.bat", "a")
        n = f + "\n"

        print('data:', data)
        print('dat:', dat)
        for j in data:
            # bat file mai jo data leya hai vo space se split keya hai so jaha space
hai vo list mai jaye
            ls = list(j.split(" "))

            if ls[0] == "gcc" or ls[0] == "g++":
                print(ls[0])
                # so ls of 1 pe sib file ka nam or extension aayega
                a = ls[1]
                b = ls[3]
                dat = dat.replace(a, str(f))
                dat = dat.replace(b, str(e[0]) + '\n')
                print('dat1' + dat)
            if (ext == '.cpp' and ls[0] == 'gcc'):
                g = ls[0]
                print('g==' + g)
                dat = dat.replace(g, "g++")
            if (ext == '.c' and ls[0] == "g++"):
                g = ls[0]
                print('g' + g)
                dat = dat.replace(g, "gcc")
            if ls[0] == "cd":
                a = ls[1]
                print('a' + a)
                # s=dat[11]+dat[12] if using deactivate
                s = dat[0] + dat[1]
                print('s' + s)
                s1 = fileloc[0] + fileloc[1]
                print('s1' + s1)
                dat = dat.replace(a, str(fileloc) + "\n")
                dat = dat.replace(f, "", 1)
                dat = dat.replace(s, s1, 1)
                print('dat3' + dat)

        ob.close()
        print(dat)
        ob = open("Resources\\c.bat", "w")
        ob.write(dat)
        ob.close()
    dig.Destroy()


else:
    print("enter in else")
    fileloc = fileloc
```

```
        print("fileloc", fileloc)
        f = name
        text = str(self.control.GetValue())
        file = open(os.path.join(fileloc), "w")
        file.write(text)
        print('f', f)
        # yaha milta hai.txt
        e = os.path.splitext(f)
        print(e)
        ext = e[1]
        print(ext)
        ch = ext
        # abhi ke leye aapne jaha bat file save ki vaha ka path dalo uska solution hai
//fileloc se hota vo mai kal
        # send karti
        if (ext == '.java'):
            ob = open("Resources\\file.bat", "r")
            dat = ob.read()
            ob.close()
            ob = open("Resources\\file.bat", "r")
            data = ob.readlines()
            ob.close()
            ob = open("Resources\\file.bat", "a")
            n = f + "\n"

            print('data:', data)
            print('dat:', dat)
            for j in data:
                # bat file mai jo data leya hai vo space se split keya hai so jaha space hai vo
list mai jaye
                ls = list(j.split(" "))
                if ls[0] == "javac":
                    print(ls[0])
                    # so ls of 1 pe sib file ka nam or extension aayega
                    a = ls[1]
                    dat = dat.replace(a, str(f) + "\n")
                    print('dat1' + dat)
                if ls[0] == "java":
                    ll = f.split(".")
                    a = ls[1]

                    dat = dat.replace(a, ll[0] + "\n")
                    print('dat2' + dat)
                if ls[0] == "cd":
                    a = ls[1]
                    print('a' + a)
                    s = dat[0] + dat[1]
                    print('s' + s)
                    s1 = fileloc[0] + fileloc[1]
                    print('s1' + s1)
                    dat = dat.replace(a, str(fileloc) + "\n")
                    dat = dat.replace(f, "", 1)
                    dat = dat.replace(s, s1, 1)
                    print('dat3' + dat)

            ob.close()
            print(dat)
            ob = open("Resources\\file.bat", "w")
            ob.write(dat)
            ob.close()

        if (ext == '.c' or ext == '.cpp'):
            print("enter in cpp")
            ob = open("Resources\\c.bat", "r")
            dat = ob.read()
            ob.close()
            ob = open("Resources\\c.bat", "r")
            data = ob.readlines()
```

```
ob.close()
#..
ob = open("Resources\\c.bat", "a")
n = f + "\n"

print('data:', data)
print('dat:', dat)
for j in data:
    # bat file mai jo data leya hai vo space se split keya hai so jaha space hai vo
list mai jaye
    ls = list(j.split(" "))

    if ls[0] == "gcc" or ls[0] == "g++":
        print(ls[0])
        # so ls of 1 pe sib file ka nam or extension aayega
        a = ls[1]
        b = ls[3]
        dat = dat.replace(a, str(f))
        dat = dat.replace(b, str(e[0]) + '\n')
        print('dat1' + dat)
    if (ext == '.cpp' and ls[0] == 'gcc'):
        g = ls[0]
        print('g==' + g)
        dat = dat.replace(g, "g++")
    if (ext == '.c' and ls[0] == "g++"):
        g = ls[0]
        print('g' + g)
        dat = dat.replace(g, "gcc")
    if ls[0] == "cd":
        a = ls[1]
        print('a' + a)
        # s=dat[11]+dat[12] if using deactivate
        s = dat[0] + dat[1]
        print('s' + s)
```

```
                s1 = fileloc[0] + fileloc[1]
                print('s1' + s1)
                dat = dat.replace(a, str(fileloc) + "\n")
                dat = dat.replace(f, "", 1)
                dat = dat.replace(s, s1, 1)
                print('dat3' + dat)

            ob.close()
            print(dat)
            ob = open("Resources\\c.bat", "w")
            ob.write(dat)
            ob.close()

    def save_as(self, e):
        global fileloc
        global name
        initial_dir = r"C:\Users\vaish"
        # the filetype mask (default is all files)
        mask = [("Text and Python files", "*.txt *.py "),
            ("C files", '*.c'),
            ("cpp files", '*.cpp'),
            ("HTML files", "*.htm"),
            ("All files", "*.*")]
        dig = wx.FileDialog(self, "Save File As", initial_dir, "Untitled", "*.*",
wx.FD_SAVE | wx.FD_OVERWRITE_PROMPT)
        if (dig.ShowModal() == wx.ID_OK):
            name = dig.GetFilename()
            dirname = dig.GetDirectory()
            text = str(self.control.GetValue())

            # location of file
            fileloc = dig.GetPath()
            print("fileloc", fileloc)
            file = open(os.path.join(dirname, name), "w")
            file.write(text)
            # path of file
            f = name
            print('f', f)
            file.close()
            # yaha milta hai.txt
            e = os.path.splitext(f)
            ext = e[1]
            print(ext)
            ch = ext
            # abhi ke leye aapne jaha bat file save ki vaha ka path dalo uska solution hai
//fileloc se hota vo mai kal
            # send karti
            if (ext == '.java'):
                ob = open("Resources\\file.bat", "r")
                dat = ob.read()
                ob.close()
                ob = open("Resources\\file.bat", "r")
                data = ob.readlines()
                ob.close()
                ob = open("Resources\\file.bat", "a")
                n = f + "\n"

                print('data:', data)
                print('dat:', dat)
                for j in data:
                    # bat file mai jo data leya hai vo space se split keya hai so jaha space hai vo
list mai jaye
                    ls = list(j.split(" "))
                    if ls[0] == "javac":
                        print(ls[0])
                        # so ls of 1 pe sib file ka nam or extension aayega
                        a = ls[1]
                        dat = dat.replace(a, str(f) + "\n")
                        print('dat1' + dat)
                    if ls[0] == "java":
                        ll = f.split(".")
                        a = ls[1]

                        dat = dat.replace(a, ll[0] + "\n")
                        print('dat2' + dat)
                    if ls[0] == "cd":
                        a = ls[1]
                        print('a' + a)
                        s = dat[0] + dat[1]
                        print('s' + s)
                        s1 = fileloc[0] + fileloc[1]
                        print('s1' + s1)
                        dat = dat.replace(a, str(fileloc) + "\n")
                        dat = dat.replace(f, "", 1)
                        dat = dat.replace(s, s1, 1)
                        print('dat3' + dat)

                ob.close()
                print(dat)
                ob = open("Resources\\file.bat", "w")
                ob.write(dat)
                ob.close()

            if (ext == '.c' or ext == '.cpp'):
                ob = open("Resources\\c.bat", "r")
                dat = ob.read()
                ob.close()
                ob = open("Resources\\c.bat", "r")
                data = ob.readlines()
                ob.close()
                ob = open("Resources\\c.bat", "a")
                n = f + "\n"
```

```
        print('data:', data)
        print('dat:', dat)
        for j in data:
            # bat file mai jo data leya hai vo space se split keya hai so jaha space hai vo
list mai jaye
            ls = list(j.split(" "))

            if ls[0] == "gcc" or ls[0] == "g++":
                print(ls[0])
                # so ls of 1 pe sib file ka nam or extension aayega
                a = ls[1]
                b = ls[3]
                dat = dat.replace(a, str(f))
                dat = dat.replace(b, str(e[0]) + '\n')
                print('dat1' + dat)
            if (ext == '.cpp' and ls[0] == 'gcc'):
                g = ls[0]
                print('g==' + g)
                dat = dat.replace(g, "g++")
            if (ext == '.c' and ls[0] == "g++"):
                g = ls[0]
                print('g' + g)
                dat = dat.replace(g, "gcc")
            if ls[0] == "cd":
                a = ls[1]
                print('a' + a)
                # s=dat[11]+dat[12] if using deactivate
                s = dat[0] + dat[1]
                print('s' + s)
                s1 = fileloc[0] + fileloc[1]
                print('s1' + s1)
                dat = dat.replace(a, str(fileloc) + "\n")
                dat = dat.replace(f, "", 1)
```

```
                        dat = dat.replace(s, s1, 1)
                        print('dat3' + dat)

                ob.close()
                print(dat)
                ob = open("Resources\\c.bat", "w")
                ob.write(dat)
                ob.close()
        dig.Destroy()

    def select(self, e):
        id = self.tree1.GetSelection()
        if id == 'root' or id == 'java' or id == 'c' or id == 'cpp' or id == 'Syntax' or id ==
'Package' or id == 'Class' or id == 'Method':
            pass
        else:
            text = self.tree1.GetItemData(id)
            self.control.AddText(text)


    def open1(self, e):
        global name
        global fileloc
        initial_dir = "C:\Temp"
        # the filetype mask (default is all files)
        mask = [("Text and Python files", "*.txt *.py "),
                ("HTML files", "*.htm"),
                ("C files", '*.c'),
                ("cpp files", '*.cpp'),
                ("All files", "*.*")]
        dig = wx.FileDialog(self, "Choose a File", initial_dir, "", "*.*", wx.FD_OPEN)
        if (dig.ShowModal() == wx.ID_OK):
            name = dig.GetFilename()
            dirname = dig.GetDirectory()
            fileloc = dig.GetPath()
            f = open(os.path.join(dirname, name), "r")
            self.control.SetValue(f.read())
        dig.Destroy()
        print("enter in else")
        fileloc = fileloc
        print("fileloc", fileloc)
        f = name
        text = str(self.control.GetValue())
        file = open(os.path.join(fileloc), "w")
        file.write(text)
        print('f', f)
        # yaha milta hai.txt
        e = os.path.splitext(f)
        print(e)
        ext = e[1]
        print(ext)
        ch = ext
        # abhi ke leye aapne jaha bat file save ki vaha ka path dalo uska solution hai //fileloc
se hota vo mai kal
        # send karti
        if (ext == '.java'):
            ob = open("Resources\\file.bat", "r")
            dat = ob.read()
            ob.close()
            ob = open("Resources\\file.bat", "r")
            data = ob.readlines()
            ob.close()
            ob = open("Resources\\file.bat", "a")
            n = f + "\n"

            print('data:', data)
            print('dat:', dat)
            for j in data:
                # bat file mai jo data leya hai vo space se split keya hai so jaha space hai vo
```

```
list mai jaye
            ls = list(j.split(" "))
            if ls[0] == "javac":
                print(ls[0])
                # so ls of 1 pe sib file ka nam or extension aayega
                a = ls[1]
                dat = dat.replace(a, str(f) + "\n")
                print('dat1' + dat)
            if ls[0] == "java":
                ll = f.split(".")
                a = ls[1]


                dat = dat.replace(a, ll[0] + "\n")
                print('dat2' + dat)
            if ls[0] == "cd":
                a = ls[1]
                print('a' + a)
                s = dat[0] + dat[1]
                print('s' + s)
                s1 = fileloc[0] + fileloc[1]
                print('s1' + s1)
                dat = dat.replace(a, str(fileloc) + "\n")
                dat = dat.replace(f, "", 1)
                dat = dat.replace(s, s1, 1)
                print('dat3' + dat)


         ob.close()
         print(dat)
         ob = open("Resources\\file.bat", "w")
         ob.write(dat)
         ob.close()


      if (ext == '.c' or ext == '.cpp'):
         print("enter in cpp")
```

```
ob = open("Resources\\c.bat", "r")
dat = ob.read()
ob.close()
ob = open("Resources\\c.bat", "r")
data = ob.readlines()
ob.close()
ob = open("Resources\\c.bat", "a")
n = f + "\n"

print('data:', data)
print('dat:', dat)
for j in data:
    # bat file mai jo data leya hai vo space se split keya hai so jaha space hai vo
list mai jaye
    ls = list(j.split(" "))

    if ls[0] == "gcc" or ls[0] == "g++":
        print(ls[0])
        # so ls of 1 pe sib file ka nam or extension aayega
        a = ls[1]
        b = ls[3]
        dat = dat.replace(a, str(f))
        dat = dat.replace(b, str(e[0]) + '\n')
        print('dat1' + dat)
    if (ext == '.cpp' and ls[0] == 'gcc'):
        g = ls[0]
        print('g==' + g)
        dat = dat.replace(g, "g++")
    if (ext == '.c' and ls[0] == "g++"):
        g = ls[0]
        print('g' + g)
        dat = dat.replace(g, "gcc")
    if ls[0] == "cd":
        a = ls[1]
```

```
                print('a' + a)
                # s=dat[11]+dat[12] if using deactivate
                s = dat[0] + dat[1]
                print('s' + s)
                s1 = fileloc[0] + fileloc[1]
                print('s1' + s1)
                dat = dat.replace(a, str(fileloc) + "\n")
                dat = dat.replace(f, "", 1)
                dat = dat.replace(s, s1, 1)
                print('dat3' + dat)

            ob.close()
            print(dat)
            ob = open("Resources\\c.bat", "w")
            ob.write(dat)
            ob.close()

    def exitmethod(self, e):
        m = wx.MessageDialog(self, "Confirm Exit", "Are you sure you want to exit?",
wx.YES_NO | wx.ICON_QUESTION)

        if m.ShowModal() == wx.ID_YES:
            m.Destroy()
        else:
            m.SetMessage("Decision", "Good Decision, continue in application")

    def newproj(self, e):
        m = wx.MessageDialog(self, "Confirm Exit", "do you want save the existing file?",
wx.YES_NO | wx.ICON_QUESTION)
        if m.ShowModal() == wx.ID_NO:
            self.control.SetValue("")
        else:
            self.save(e)
            self.control.SetValue("")
```

```
def OnCut(self, e):
    self.control.Cut()

def OnCopy(self, e):
    self.control.Copy()

def OnPaste(self, e):
    self.control.Paste()

def OnRedo(self, e):
    self.control.Redo()

def OnUndo(self, e):
    self.control.Undo()

def run(self, e):
    global name
    r = r"Resources\\"
    n = "file.bat"
    r = r + n
    os.startfile(r)

def runc(self, e):
    global name
    r = r"Resources\\"
    n = "c.bat"
    r = r + n
    os.startfile(r)

def tts(self, e):
    text = str(self.control.GetValue())
    b = text.replace("{", "open brace")
    c = b.replace("}", "close brace")
```

```
        d = c.replace("\n", " ")
        e = d.replace(":", "colon")
        f = e.replace(";", "semicolon")
        g = f.replace("(", "open parentheses")
        h = g.replace(")", "close parentheses")
        ee=pyttsx3.init()
        rare =ee.getProperty("rate")
        ee.setProperty("rate",100)
        volum=ee.getProperty("volume")
        ee.setProperty("volume",3.0)
        voice=ee.getProperty("voices")
        ee.setProperty("voice",voice[1].id)
        ee.say(h)
        ee.runAndWait()
    def OnToggleLineNumbers(self, e):
        if (self.lineNumbersEnabled):
            self.control.SetMarginWidth(1, 0)
            self.lineNumbersEnabled = False
        else:
            self.control.SetMarginWidth(1, self.leftMarginWidth)
            self.lineNumbersEnabled = True

    def Fontbox(self, e):
        fontdialog = wx.FontDialog(self)
        fontdialog.CenterOnParent();
        if fontdialog.ShowModal() == wx.ID_OK:
            data = fontdialog.GetFontData()
            myFont = data.GetChosenFont()
            self.control.StyleSetFont(0, font=myFont)
        fontdialog.Destroy()

    def UpdateLineCol(self, e):
        line = self.control.GetCurrentLine() + 1
        col = self.control.GetColumn(self.control.GetCurrentPos())
        stat = "Line %s, Column %s" % (line, col)
        self.StatusBar.SetStatusText(stat, 0)

    def SpeechDef(self, e):
        text = str(self.control.GetValue())
        b = text.replace("{", "open brace")
        c = b.replace("}", "close brace")
        d = c.replace("\n", " ")
        e = d.replace(":", "colon")
        f = e.replace(";", "semicolon")
        g = f.replace("(", "open parentheses")
        h = g.replace(")", "close parentheses")
        e = pyttsx3.init()
        rare = e.getProperty("rate")
        e.setProperty("rate", 100)
        volum = e.getProperty("volume")
        e.setProperty("volume", 3.0)
        voice = e.getProperty("voices")
        e.setProperty("voice", voice[0].id)
        e.say(h)
        e.runAndWait()



    def OnLeftUp(self, e):
        # This way if you click on another position in the text box
        # it will update the line/col number in the status bar (like it should)
        self.UpdateLineCol(self)
        e.Skip()
    def howto(self,e):
        print("file")
        r=r"Resources\\"
        n='manual.pdf'
        r=r+n
        os.startfile(r)
    def OnAbout(self, e):
        # Simple display a modal window telling about the application
        dlg = wx.MessageDialog(self,"An elegant, yet simple, text editor made with
Python and wxPython.\n\nCreated by:\n"
                              "\n1. Vaishnavi Nighvekar"
                              "\n2. Om Agrawal"
                              "\n3. Prakash Zodge"
                              "\n3. Yash Jaiswal"
                              "\n\nVersion 1.0.0\n\n",
                          "About Limitless Editor", wx.OK)
        dlg.ShowModal()
        dlg.Destroy()


app = wx.App(False)
Demo = Editor(None, "Limitless Editor")
Demo.SetIcon(wx.Icon('Resources/le.png'))
app.MainLoop()
```

8.2 CODE OF TREE:-

```
import wx

class Tree(wx.TreeCtrl):
    def __init__(self,parent,style):
        #Initialize the tree
        wx.TreeCtrl.__init__(self ,parent ,style)



        root = self.AddRoot('LEMITLESS_TREE')
        java = self.AppendItem(root,'Java')
        c = self.AppendItem(root,'C')
        cpp = self.AppendItem(root,'CPP')
        image_list = wx.ImageList(16,16)
        LEMITLESS_TREE=image_list.Add(wx.Image("Resources/dd.png",
wx.BITMAP_TYPE_PNG).Scale(16, 16).ConvertToBitmap())
        Java            =           image_list.Add(wx.Image("Resources/java.png",
wx.BITMAP_TYPE_PNG).Scale(16, 16).ConvertToBitmap())
        C               =              image_list.Add(wx.Image("Resources/c.png",
wx.BITMAP_TYPE_PNG).Scale(16, 16).ConvertToBitmap())
        CPP=                          image_list.Add(wx.Image("Resources/cpp.png",
wx.BITMAP_TYPE_PNG).Scale(16, 16).ConvertToBitmap())
        self.AssignImageList(image_list)
        self.SetItemImage(root,LEMITLESS_TREE, wx.TreeItemIcon_Normal)
        self.SetItemImage(java,Java, wx.TreeItemIcon_Normal)
        self.SetItemImage(c,C, wx.TreeItemIcon_Normal)
        self.SetItemImage(cpp,CPP, wx.TreeItemIcon_Normal)

        #Tree for Java

        Package = self.AppendItem(java,text='Package',data='')

        #Class = self.AppendItem(java,'Class')
        #Method = self.AppendItem(java,'Method')
        #AwtControls = self.AppendItem(java,'Awt Controls')

        #Packages for Java

        self.AppendItem(Package,text='io',data='import java.io.*;')
        self.AppendItem(Package,text='util',data='import java.util.*;')
        self.AppendItem(Package ,text='awt',data='import java.awt.*;')
        self.AppendItem(Package, text='net', data='import java.net.*;')
        self.AppendItem(Package, text='event', data='import java.awt.event;')
        self.AppendItem(Package, text='lang', data='import java.lang.*;')
        self.AppendItem(Package, text='applet', data='import java.applet.*;')
        self.AppendItem(Package, text='swing', data='import javax.swing.*;')
        self.AppendItem(Package, text='swingEvent', data='import javax.swing.event;')
        self.AppendItem(Package, text='servlet', data='import javax.servlet.*;')
        self.AppendItem(Package, text='servletJsp', data='import javax.servlet.jsp;')

        #Loops of Java
        Loops = self.AppendItem(java,text='Loops',data='')
        self.AppendItem(Loops,text='while()',data='while(condition)\n{\n  \n}')
        self.AppendItem(Loops, text='for()', data='for(condition)\n{\n  \n}')
        self.AppendItem(Loops, text='do()', data='do\n{\n  \n}while(condition);')

        Conditional = self.AppendItem(java, text='Conditional and Return Statements',
data='')
        self.AppendItem(Conditional, text='if()', data='if(condition)\n{\n  \n}')
        self.AppendItem(Conditional, text='if else()', data='if(condition)\n{\n
\n}\nelse\n{\n  \n}')
        self.AppendItem(Conditional, text='else if()', data='if(condition)\n{\n   \n}\nelse
if(condition)\n{\n  \n}\nelse\n{\n  \n}')
        self.AppendItem(Conditional, text='try catch', data='try\n{\n
\n}\ncatch(Exception e)\n{\n  \n}')
        self.AppendItem(Conditional, text='try catch finally', data='try\n{\n
\n}\ncatch(Exception e)\n{\n  \n}\nfinally\n{\n  \n}')
```

```
Functions = self.AppendItem(java, text='Functions', data='')

StringFunctions = self.AppendItem(Functions, text='String Functions', data='')
self.AppendItem(StringFunctions, text='compareTo()', data='compareTo()')
self.AppendItem(StringFunctions, text='equals()', data='equals()')
self.AppendItem(StringFunctions, text='valueOf()', data='valueOf()')
self.AppendItem(StringFunctions, text='substring()', data='substring()')
self.AppendItem(StringFunctions, text='replace()', data='replace()')
self.AppendItem(StringFunctions, text='toString()', data='toString()')
self.AppendItem(StringFunctions, text='charAt()', data='charAt()')
self.AppendItem(StringFunctions, text='concat()', data='concat()')
self.AppendItem(StringFunctions, text='endsWith()', data='endsWith()')
self.AppendItem(StringFunctions, text='lastIndexOf()', data='lastIndexOf()')
self.AppendItem(StringFunctions, text='firstIndexOf()', data='firstIndexOf()')
self.AppendItem(StringFunctions, text='getChars()', data='getChars()')
self.AppendItem(StringFunctions, text='getBytes()', data='getBytes()')
self.AppendItem(StringFunctions, text='indexOf()', data='indexOf()')
self.AppendItem(StringFunctions, text='length()', data='length()')
self.AppendItem(StringFunctions, text='split()', data='split()')
self.AppendItem(StringFunctions, text='startsWith()', data='startsWith()')
self.AppendItem(StringFunctions, text='toLowerCase()', data='toLowerCase()')
self.AppendItem(StringFunctions, text='toUppercase()', data='toUpperCase()')

MathFunctions = self.AppendItem(Functions, text='Math Functions', data='')
self.AppendItem(MathFunctions, text='pow()', data='pow()')
self.AppendItem(MathFunctions, text='sqrt()', data='sqrt()')
self.AppendItem(MathFunctions, text='abs()', data='abs()')
self.AppendItem(MathFunctions, text='ceil()', data='ceil()')
self.AppendItem(MathFunctions, text='floor()', data='floor()')
self.AppendItem(MathFunctions, text='floorDiv()', data='floorDiv()')
self.AppendItem(MathFunctions, text='min()', data='min()')
self.AppendItem(MathFunctions, text='max()', data='max()')
self.AppendItem(MathFunctions, text='round()', data='round()')
self.AppendItem(MathFunctions, text='random()', data='random()')
```

```
        ParsingFunctions = self.AppendItem(Functions, text='Parsing Functions', data='')
        self.AppendItem(ParsingFunctions,                         text='Integer.parseInt()',
data='Integer.parseInt()')
        self.AppendItem(ParsingFunctions,                         text='Float.parseFloat()',
data='Float.parseFloat()')
        self.AppendItem(ParsingFunctions,                      text='Double.parseDouble()',
data='Double.parseDouble()')
        self.AppendItem(ParsingFunctions,                          text='Byte.praseByte()',
data='Byte.praseByte()')

        AWTFunctions = self.AppendItem(Functions, text='Parsing Functions', data='')
        self.AppendItem(AWTFunctions, text='setSize()', data='setSize()')
        self.AppendItem(AWTFunctions, text='setVisible()', data='setVisible()')
        self.AppendItem(AWTFunctions, text='setTitle()', data='setTitle()')
        self.AppendItem(AWTFunctions, text='setText()', data='setText()')
        self.AppendItem(AWTFunctions, text='setAlignment()', data='setAlignment()')
        self.AppendItem(AWTFunctions, text='getAlignment()', data='getAlignment()')
        self.AppendItem(AWTFunctions, text='setLabel()', data='setLabel()')
        self.AppendItem(AWTFunctions, text='getLabel()', data='getLabel()')
        self.AppendItem(AWTFunctions, text='setState()', data='setState()')
        self.AppendItem(AWTFunctions, text='getState()', data='getText()')
        self.AppendItem(AWTFunctions,                    text='getSelectedCheckBox()',
data='getSelectedCheckBox()')
        self.AppendItem(AWTFunctions,                    text='setSelectedCheckBox()',
data='setSelectedCheckBox()')
        self.AppendItem(AWTFunctions, text='add()', data='add()')
        self.AppendItem(AWTFunctions,                         text='getSelectedItem()',
data='getSelectedItem()')
        self.AppendItem(AWTFunctions,                        text='getSelectedItems()',
data='getSelectedItems()')
        self.AppendItem(AWTFunctions, text='select()', data='select()')
        self.AppendItem(AWTFunctions, text='getItemCount()', data='getItemCount()')
        self.AppendItem(AWTFunctions, text='getItem()', data='getItem()')
        self.AppendItem(AWTFunctions, text='getSelectedText()', data='getSelectedText()')
        self.AppendItem(AWTFunctions, text='isEditable()', data='isEditable()')
        self.AppendItem(AWTFunctions, text='apped()', data='apped()')
        self.AppendItem(AWTFunctions, text='insert()', data='insert()')
        self.AppendItem(AWTFunctions, text='getEchoChar()', data='getEchoChar()')
        self.AppendItem(AWTFunctions, text='setEchoChar()', data='setEchoChar()')
        self.AppendItem(AWTFunctions, text='replaceRange()', data='replaceRange()')

        AppletFunctions = self.AppendItem(Functions, text='Applet Life Cycle Methods', data='')
        self.AppendItem(AppletFunctions, text='init()', data='public void init()')
        self.AppendItem(AppletFunctions, text='start()', data='public void start()')
        self.AppendItem(AppletFunctions, text='paint()', data='public void paint()')
        self.AppendItem(AppletFunctions, text='stop()', data='public void stop()')
        self.AppendItem(AppletFunctions, text='destroy()', data='public void destroy()')
        self.AppendItem(AppletFunctions, text='repaint()', data='repaint()')

        OtherFunctions = self.AppendItem(Functions, text='Other Functions', data='')
        self.AppendItem(OtherFunctions, text='println()', data='System.out.println();')
        self.AppendItem(OtherFunctions, text='main()', data='public static void main(String ar[])\n{\n  \n}')
        self.AppendItem(AppletFunctions, text='finalize()', data='finalize()')

        AWTandSwingComponents = self.AppendItem(java, text='AWT and Swing Components', data='')
        self.AppendItem(AWTandSwingComponents, text='Label()', data='Label l = new Label("");\n')
        self.AppendItem(AWTandSwingComponents, text='Button()', data='Button b = new Button("");\n')
        self.AppendItem(AWTandSwingComponents, text='Choice()', data='Choice c = new Choice();\n')
        self.AppendItem(AWTandSwingComponents, text='TextField()', data='TextField t = new TextField("");\n')
```

self.AppendItem(AWTandSwingComponents, text='TextArea()', data='TextArea ta = new TextArea();\n')

self.AppendItem(AWTandSwingComponents, text='Checkbox()', data='Checkbox c = new Checkbox();\n')

self.AppendItem(AWTandSwingComponents, text='CheckBoxGroup()', data='Checkbox c = new Checkbox(String label, boolean state, CheckboxGroup cbg);\n')

self.AppendItem(AWTandSwingComponents, text='List()', data='List s = new List();\n')

self.AppendItem(AWTandSwingComponents, text='MenuBar()', data='MenuBar mb = new MenuBar();\n')

self.AppendItem(AWTandSwingComponents, text='Menu()', data='Menu m = new Menu("");\n')

self.AppendItem(AWTandSwingComponents, text='MenuItem()', data='MenuItem mi = new MenuItem("");\n')

self.AppendItem(AWTandSwingComponents, text='JLabel()', data='JLabel l = new JLabel("");\n')

self.AppendItem(AWTandSwingComponents, text='JButton()', data='JButton b = new JButton("");\n')

self.AppendItem(AWTandSwingComponents, text='JTextField()', data='JTextField t = new JTextField("");\n')

self.AppendItem(AWTandSwingComponents, text='JTextArea()', data='JTextArea t = new JTextArea("");\n')

self.AppendItem(AWTandSwingComponents, text='JComboBox()', data='JComboBox C = new JComboBox();\n')

self.AppendItem(AWTandSwingComponents, text='JCheckBox()', data='JCheckBox c = new JCheckBox("1");\n')

self.AppendItem(AWTandSwingComponents, text='JRadioButton()', data='JRadioButton r = new JRadioButton("1");\n')

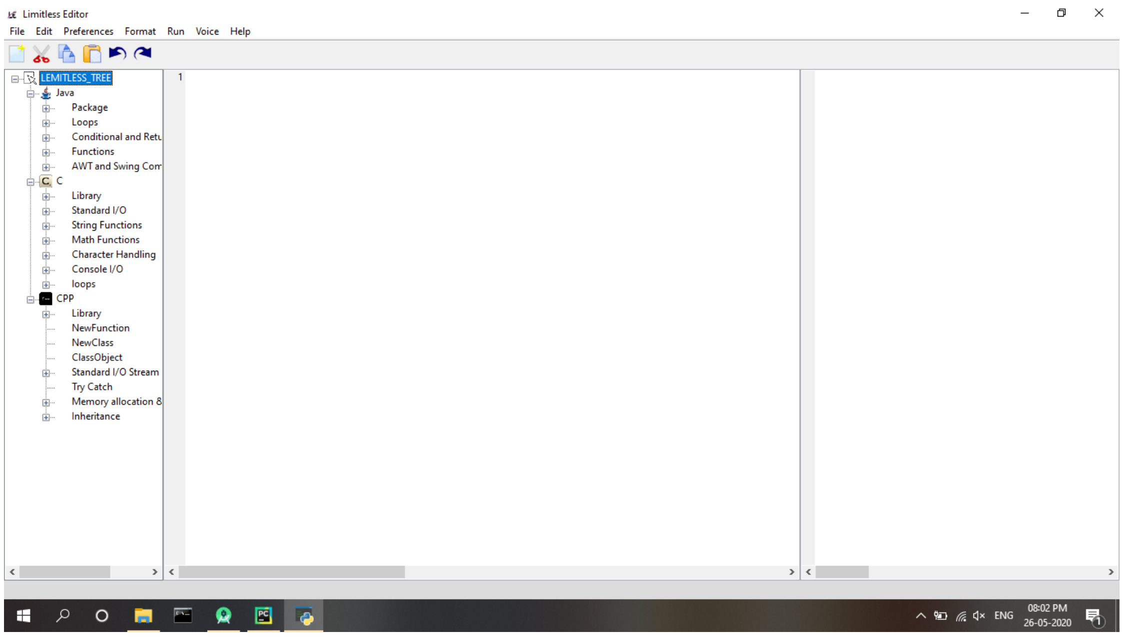
self.AppendItem(AWTandSwingComponents, text='JTree()', data='JTree jt=new JTree(DefaultMutableTreeNodeObjectName);\n')

self.AppendItem(AWTandSwingComponents, text='JPasswordField()', data='JPasswordField p = new JPasswordField();\n')

self.AppendItem(AWTandSwingComponents, text='JMenuBar()',

```
data='JMenuBar mb = new JMenuBar();\n')
        self.AppendItem(AWTandSwingComponents, text='JMenu()', data='JMenu m = new JMenu("");\n')
        self.AppendItem(AWTandSwingComponents, text='JMenuItem()', data='JMenuItem mi = new JMenuItem("");\n')

        #Tree for C
        libc = self.AppendItem(c,text='Library',data='')
        self.AppendItem(libc,text='stdio',data = "#include<stdio.h>")
        self.AppendItem(libc,text='conio',data = "#include<conio.h>")
        self.AppendItem(libc,text='string',data = "#include<string.h>")
        self.AppendItem(libc,text='stdlib',data = "#include<stdlib.h>")
        self.AppendItem(libc,text='math',data = "#include<math.h>")
        self.AppendItem(libc,text='time',data = "#include<time.h>")
        self.AppendItem(libc,text='ctype',data = "#include<ctype.h>")
        self.AppendItem(libc,text='stdarg',data = "#include<stdarg.h>")
        self.AppendItem(libc,text='signal',data = "#include<signal.h>")
        self.AppendItem(libc,text='setjmp',data = "#include<setjmp.h>")
        self.AppendItem(libc,text='locale',data = "#include<locale.h>")
        self.AppendItem(libc,text='errno',data = "#include<errno.h>")
        self.AppendItem(libc,text='assert',data = "#include<assert.h>")

        io = self.AppendItem(c,text="Standard I/O",data='')
        self.AppendItem(io,text="printf()",data="printf("")")
        self.AppendItem(io,text="scanf()",data="scanf("")")
        self.AppendItem(io,text="getchar()",data="getchar()")
        self.AppendItem(io,text="putchar()",data="putchar()")
        self.AppendItem(io,text="fopen()",data="fopen()")
        self.AppendItem(io,text="fclose()",data="fclose()")

        sf = self.AppendItem(c,text="String Functions",data='')
        self.AppendItem(sf,text = "strcmp()",data="strcmp()")
        self.AppendItem(sf,text = "strcpy()",data="strcpy()")
        self.AppendItem(sf,text = "strcat()",data="strcat()")
```

```
mf = self.AppendItem(c,text = "Math Functions",data='')
self.AppendItem(mf,text="floor()",data="floor(double x)")
self.AppendItem(mf,text="ceil()",data="ceil(double x)")
self.AppendItem(mf,text="exp()",data="exp(double x)")
self.AppendItem(mf,text="sqrt()",data="sqrt(double x)")
self.AppendItem(mf,text="pow()",data="pow(int base,int power)")
self.AppendItem(mf,text="fabs()",data="fabs(int x)")
self.AppendItem(mf,text="log()",data="log(double x)")
self.AppendItem(mf,text="sin()",data="sin(double x)")
self.AppendItem(mf,text="cos()",data="cos(double x)")
self.AppendItem(mf,text="tan()",data="tan(double x)")

ch = self.AppendItem(c,text="Character Handling",data='')
self.AppendItem(ch,text= "isalpha()",data="isalpha()")
self.AppendItem(ch,text= "isdigit()",data="isdigit()")
self.AppendItem(ch,text= "isalnum()",data="isalnum()")
self.AppendItem(ch,text= "isupper()",data="isupper()")
self.AppendItem(ch,text= "islower()",data="islower()")
self.AppendItem(ch,text= "toupper()",data="tolower()")
self.AppendItem(ch,text= "tolower()",data="tolower()")
self.AppendItem(ch,text= "iscntrl()",data="iscntrl()")
self.AppendItem(ch,text= "isgraph()",data="isgraph()")
self.AppendItem(ch,text= "isprint()",data="isprint()")
self.AppendItem(ch,text= "ispunct()",data="ispunct()")
self.AppendItem(ch,text= "isspace()",data="isspace()")
self.AppendItem(ch,text= "isxdigit()",data="isxdigit()")

cio = self.AppendItem(c,text = "Console I/O",data ="")
self.AppendItem(cio,text="clrscr()",data="clrscr()")
self.AppendItem(cio,text="getch()",data="getch()")

loops = self.AppendItem(c,text = 'loops',data = "")
self.AppendItem(loops,text="while",data            =            'while(Condition)\n{\n
Statement(s)\n    Increment/Decrement; \n}\n')
        self.AppendItem(loops,text = "for",data = 'for(initialization; condition; incOrDec)\n{\n   Statement(s) \n}\n')
        self.AppendItem(loops,text="do-while",data = 'do\n{\n         Statement(s)\n Increment/Decrement;\n}while(Condition); \n')

        # Tree For CPP
        libcpp = self.AppendItem(cpp,text = 'Library',data = '')
        self.AppendItem(libcpp,text = 'cmath',data = '#include<cmath.h>')
        self.AppendItem(libcpp,text = 'iostream',data = '#include<iostream.h>')
        self.AppendItem(libcpp,text = 'cstring',data = '#include<cstring.h>')
        self.AppendItem(libcpp,text = 'cctype',data = '#include<cctype.h>')
        self.AppendItem(libcpp,text = 'csignal',data = '#include<csignal.h>')
        self.AppendItem(libcpp,text = 'clocale',data = '#include<clocale.h>')
        self.AppendItem(libcpp,text = 'cwctype',data = '#include<cwctype.h>')
        self.AppendItem(libcpp,text = 'cstdio',data = '#include<cstdio.h>')
        self.AppendItem(libcpp,text = 'cwchar',data = '#include<cwchar.h>')
        self.AppendItem(libcpp,text = 'cuchar',data = '#include<cuchar.h>')
        self.AppendItem(libcpp,text = 'csetjmp',data = '#include<csetjmp.h>')
        self.AppendItem(libcpp,text = 'cfenv',data = '#include<cfenv.h>')
        self.AppendItem(libcpp,text = 'ctime',data = '#include<ctime.h>')

        cppFunctions = self.AppendItem(cpp, text='NewFunction',
                                data='void myFunction() \n{\n  // code to be executed\n}')
        cppClass = self.AppendItem(cpp, text='NewClass', data='class className\n{\n //Code\n};')
        cppObject = self.AppendItem(cpp, text='ClassObject', data='className ObjName = new className();')
        cppIO = self.AppendItem(cpp, text='Standard I/O Stream', data='')
        self.AppendItem(cppIO, text='Output Stream', data='cout<< ;')
        self.AppendItem(cppIO, text='Input Stream', data='cin>> ;')
        self.AppendItem(cppIO, text='Un-buffered error Stream', data='cerr << ;')
        self.AppendItem(cppIO, text='Buffered error Stream', data='clog << ;')
        cppException = self.AppendItem(cpp, text='Try Catch',
```



```
data='try\n{\n  //try code\n}\ncatch(ExceptionHere)\n{\n  //catch code\n}')
        cppMemory = self.AppendItem(cpp, text='Memory allocation & de-allocation', data='')
        self.AppendItem(cppMemory, text='New Operator', data='pointer-variable = new data-type;')
        self.AppendItem(cppMemory, text='Delete Operator',
                        data='// Release memory pointed by pointer-variable \ndelete pointer-variable; ')
        cppInheritance = self.AppendItem(cpp, text='Inheritance', data='')
        self.AppendItem(cppInheritance, text='Single Inheritance',
                        data='class subclass_name : access_mode base_class_name\n{\n  //body of subclass\n};')
        self.AppendItem(cppInheritance, text='Multiple Inheritance',
                        data='class subclass_name : access_mode base_class1, access_mode base_class2, ....\n{\n  //body of subclass\n};')
```

8.3 JAVA BATCH FILE (FILE.BAT)

```
G:
cd G:\javapro\
javac Filename.java
java Filename
pause
```

8.4 C AND C++ BATCH FILE (C.BAT)

```
G:
cd G:\
g++ Filename.cpp -o filename
Filename
Pause
```

CHAPTER 9

IMPLEMENTATION

9.1 OUTPUT OF LIMITLESS EDITOR

Figure 9.1: Start Page and Tree Structure

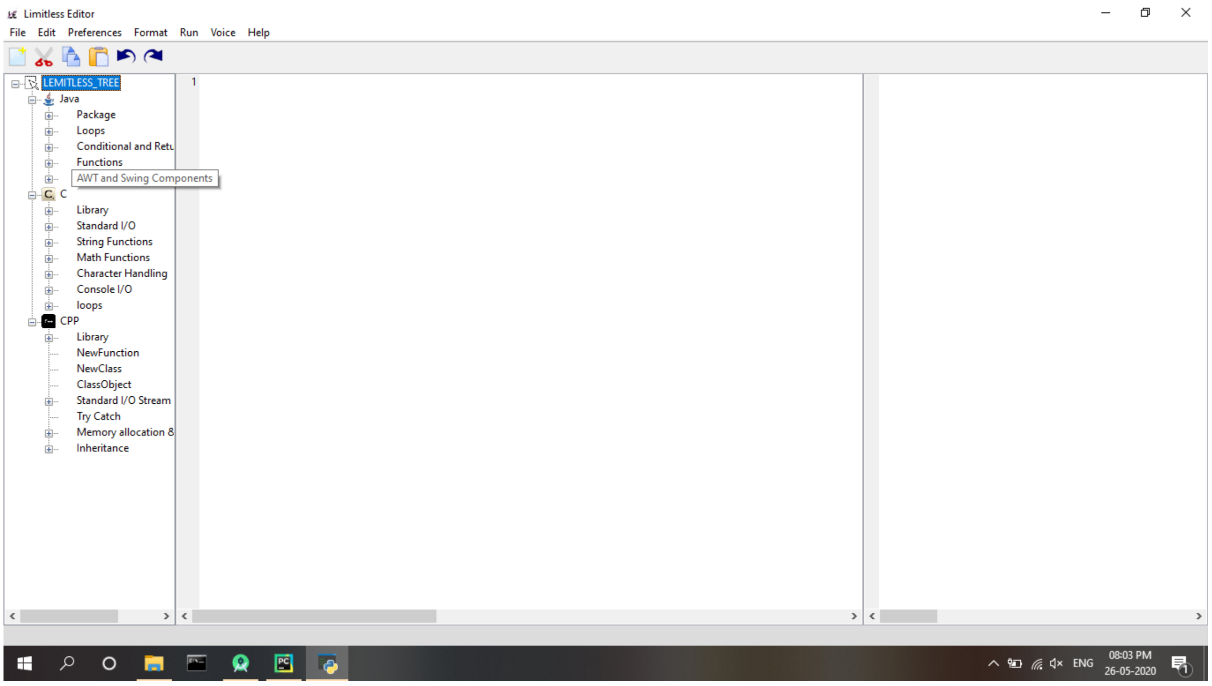


Figure 9.2: Tree Complete Pop-up

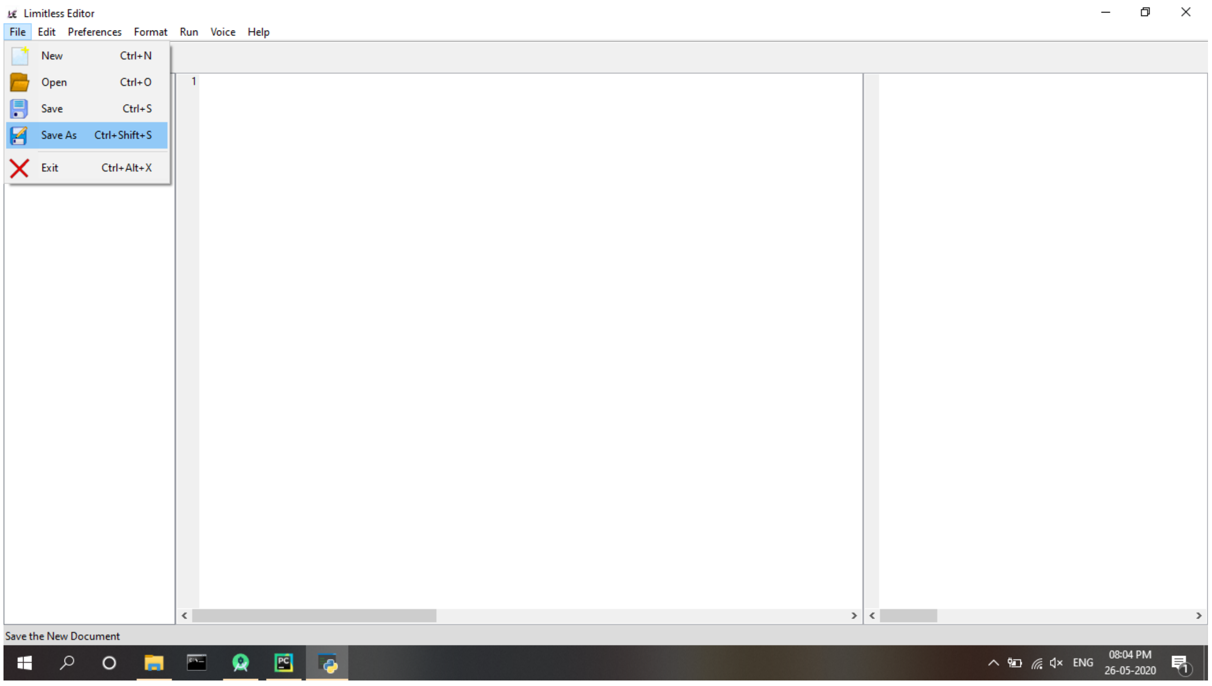


Figure 9.3: File Menu

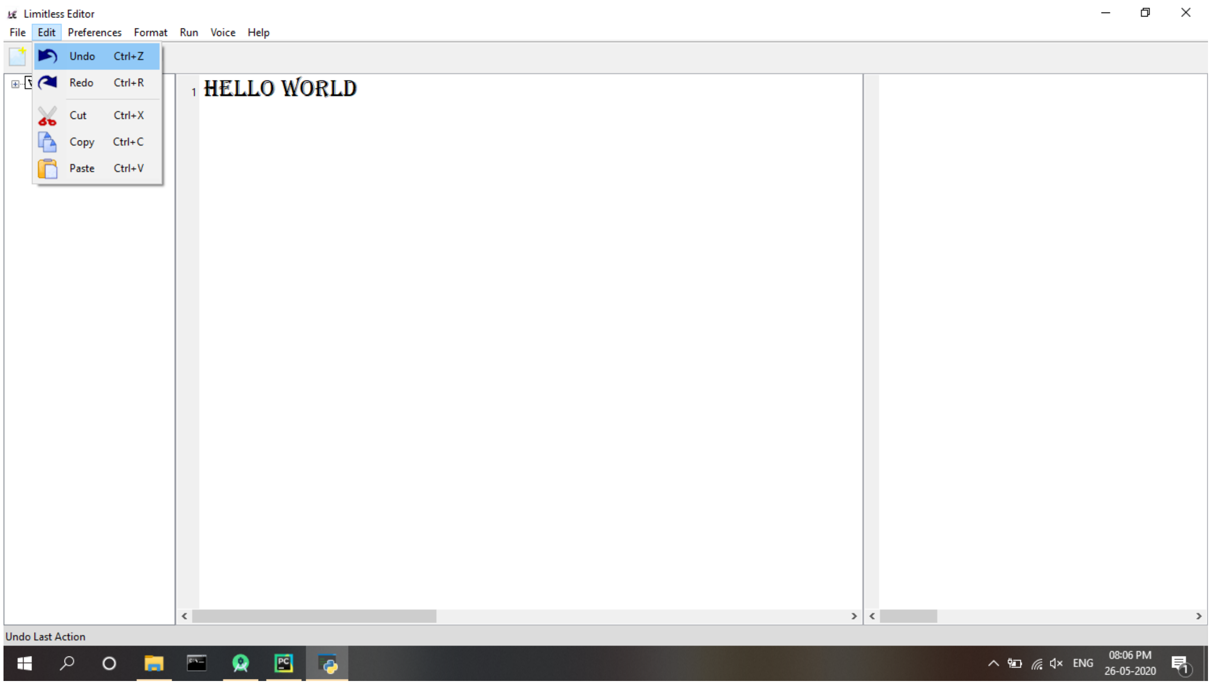
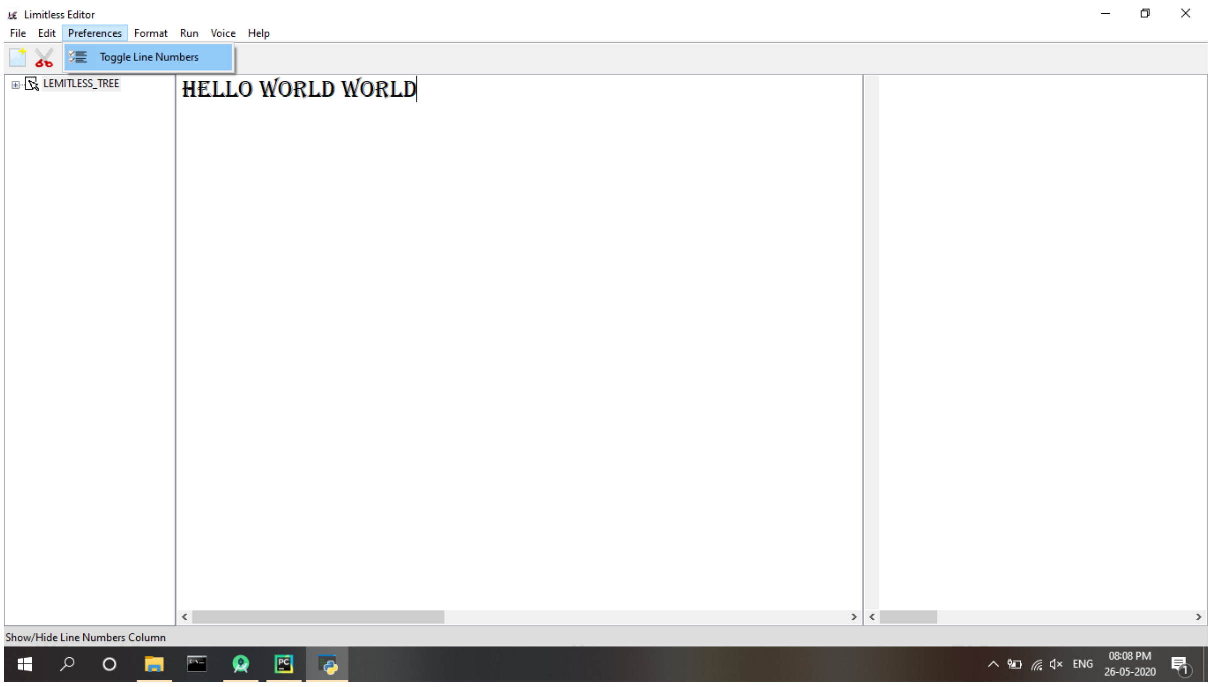


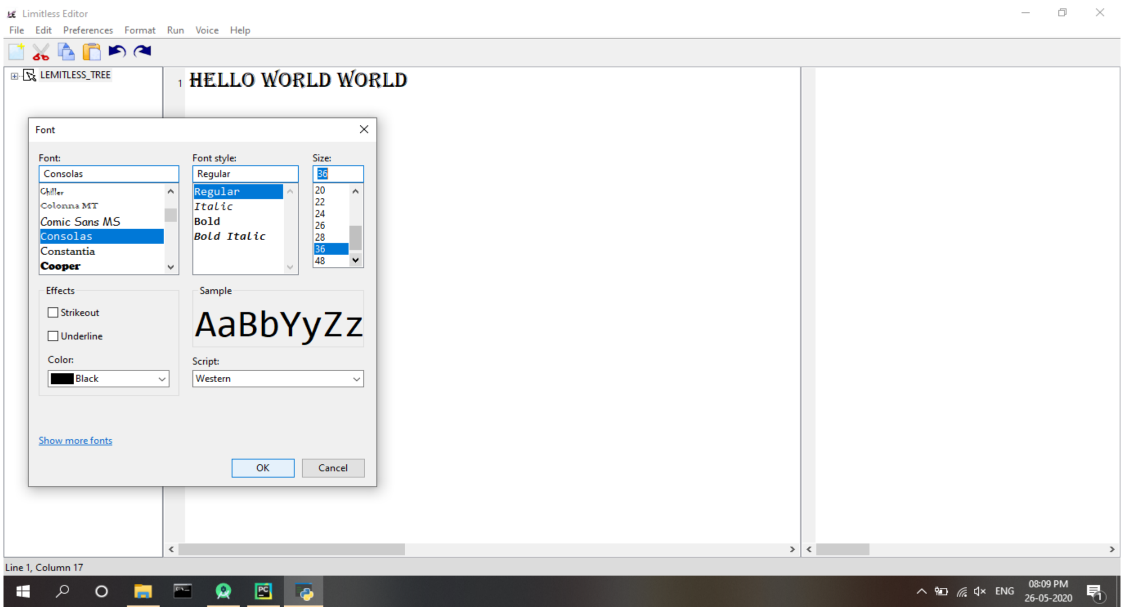
Figure 9.4: Edit Menu



HELLO WORLD WORLD

Show/Hide Line Numbers Column

Figure 9.5: Preferences> Toggle Line Numbers Menu>Line Number Hidden

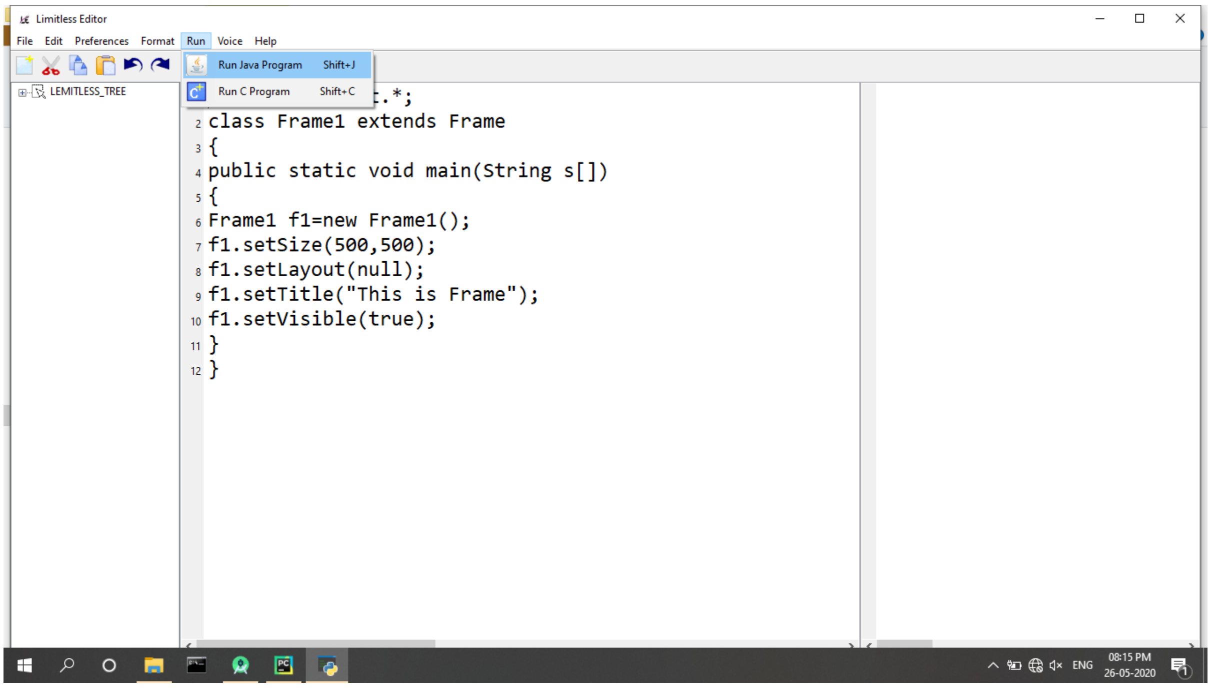


1 HELLO WORLD WORLD

Font: Consolas
Font style: Regular
Size: 36
Color: Black
Script: Western

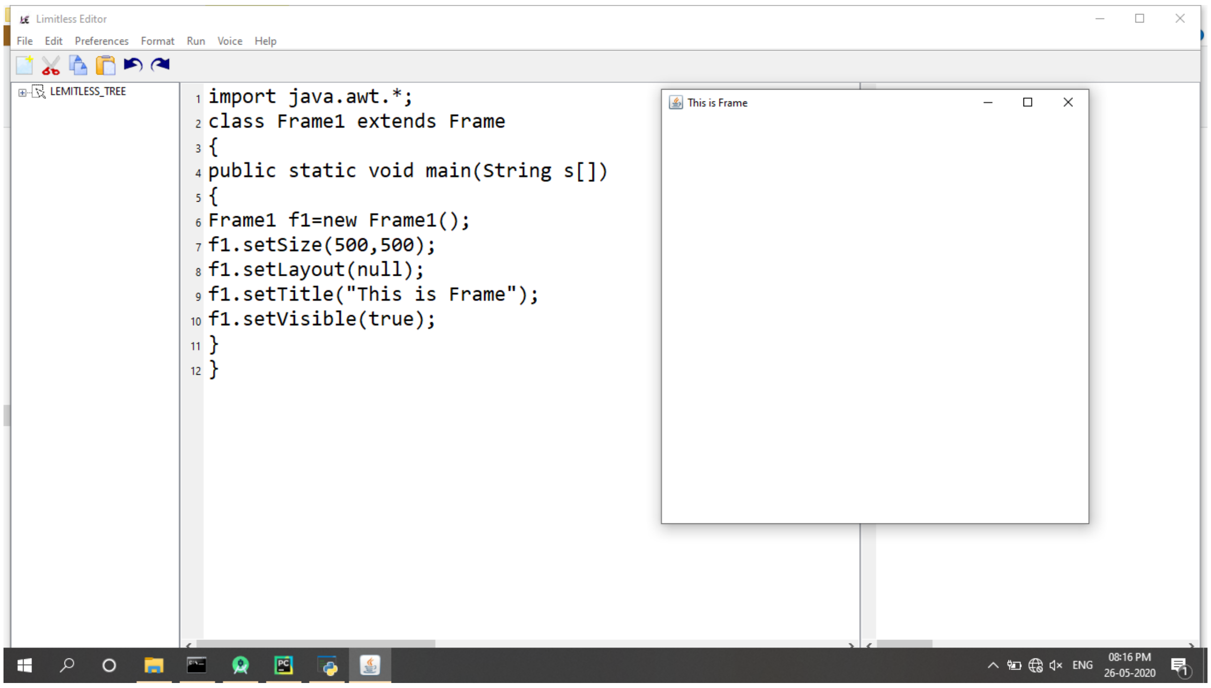
Line 1, Column 17

Figure 9.6: Format>Font Menu



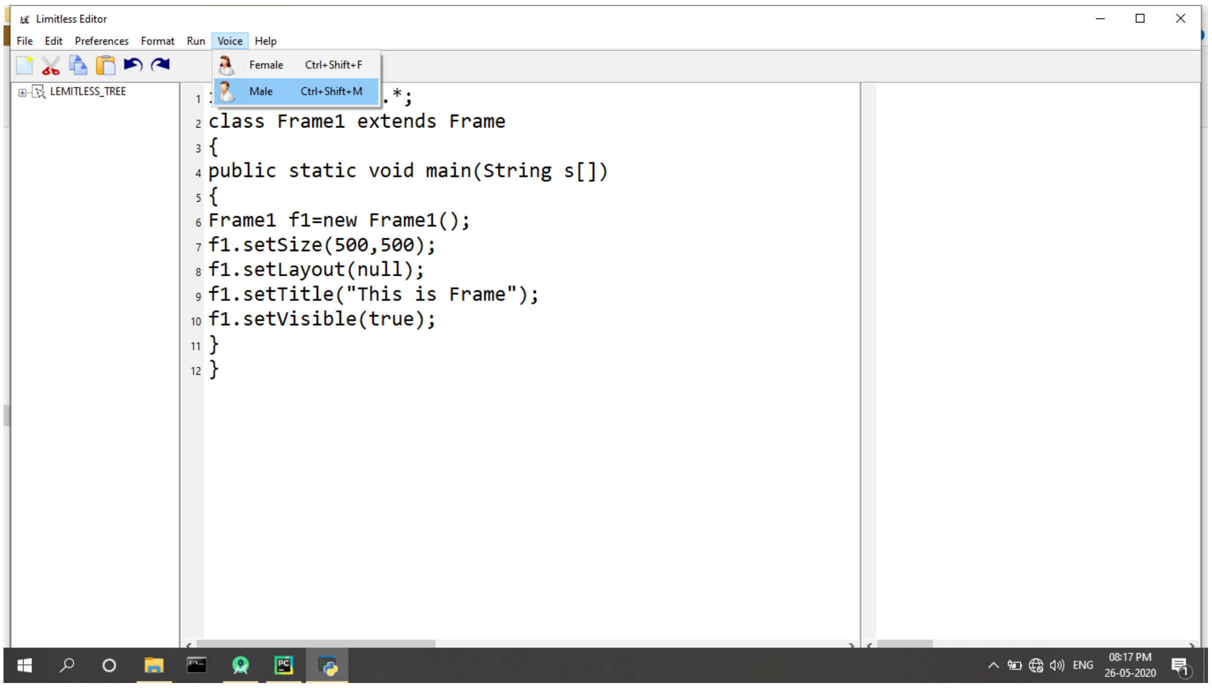
```
.*;
class Frame1 extends Frame
{
public static void main(String s[])
{
Frame1 f1=new Frame1();
f1.setSize(500,500);
f1.setLayout(null);
f1.setTitle("This is Frame");
f1.setVisible(true);
}
}
```

Figure 9.7: Run>Run Java Program Menu



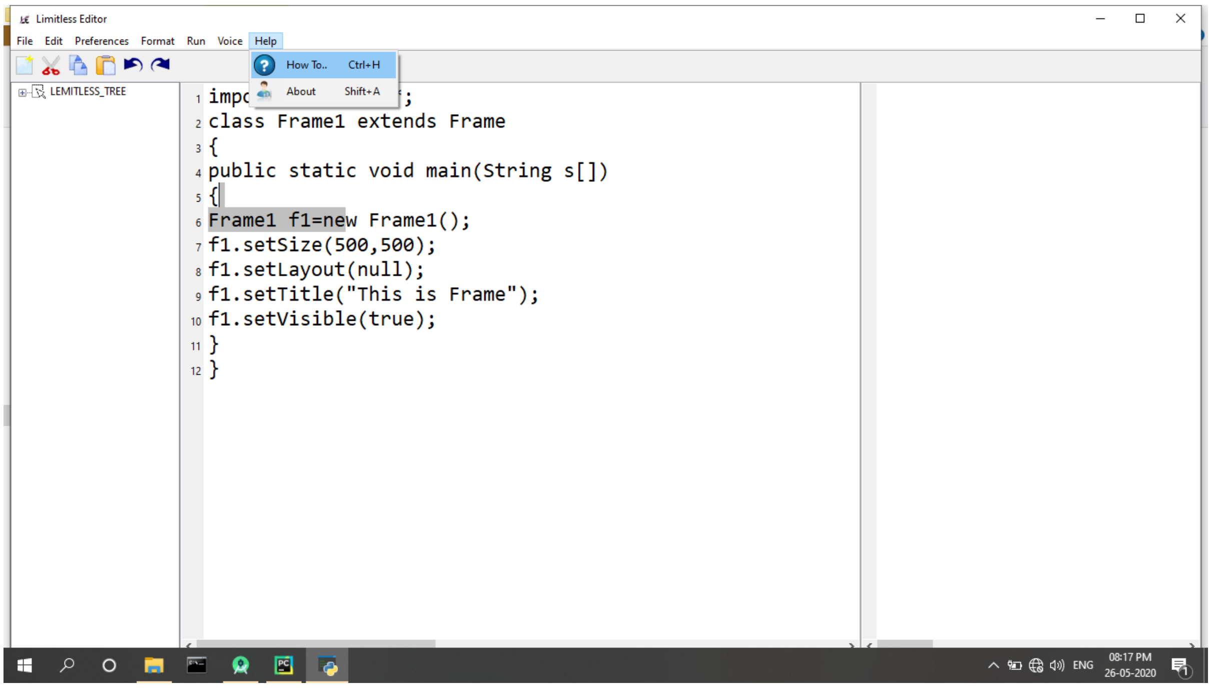
```
import java.awt.*;
class Frame1 extends Frame
{
public static void main(String s[])
{
Frame1 f1=new Frame1();
f1.setSize(500,500);
f1.setLayout(null);
f1.setTitle("This is Frame");
f1.setVisible(true);
}
}
```

Figure 9.8: JAVA Program Output



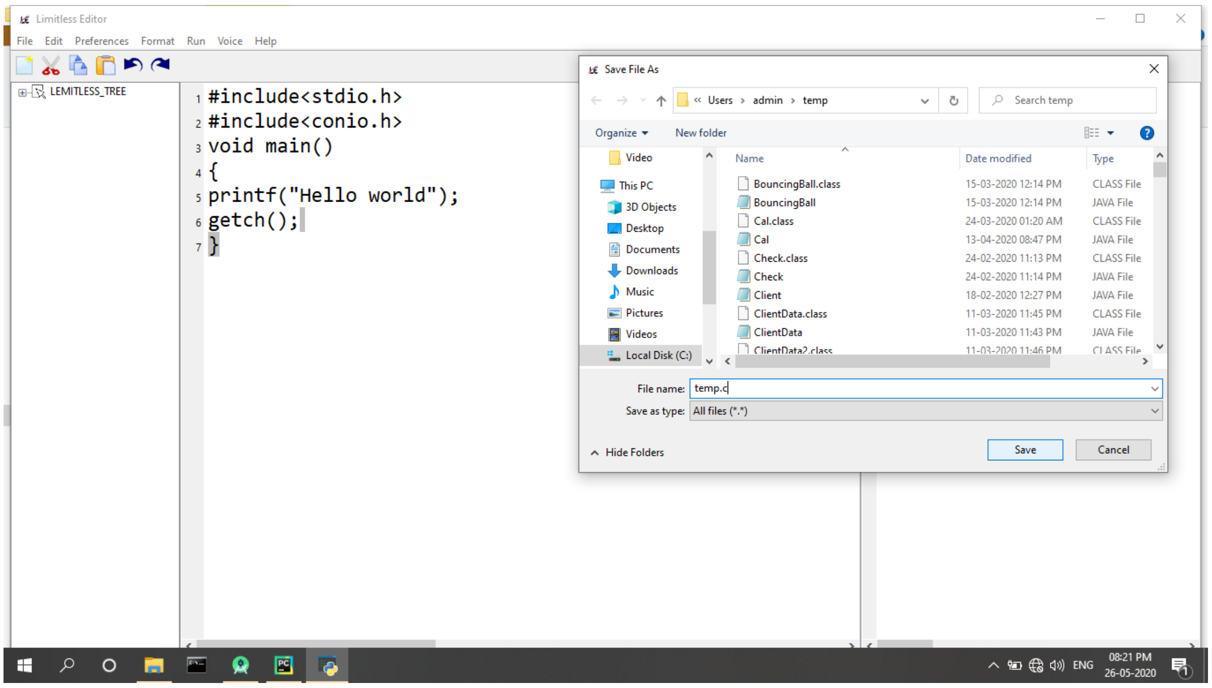
```
:              .*;
class Frame1 extends Frame
{
public static void main(String s[])
{
Frame1 f1=new Frame1();
f1.setSize(500,500);
f1.setLayout(null);
f1.setTitle("This is Frame");
f1.setVisible(true);
}
}
```

Figure 9.9: Voice



```
impo                 ;
class Frame1 extends Frame
{
public static void main(String s[])
{
Frame1 f1=new Frame1();
f1.setSize(500,500);
f1.setLayout(null);
f1.setTitle("This is Frame");
f1.setVisible(true);
}
}
```

Figure 9.10: Help

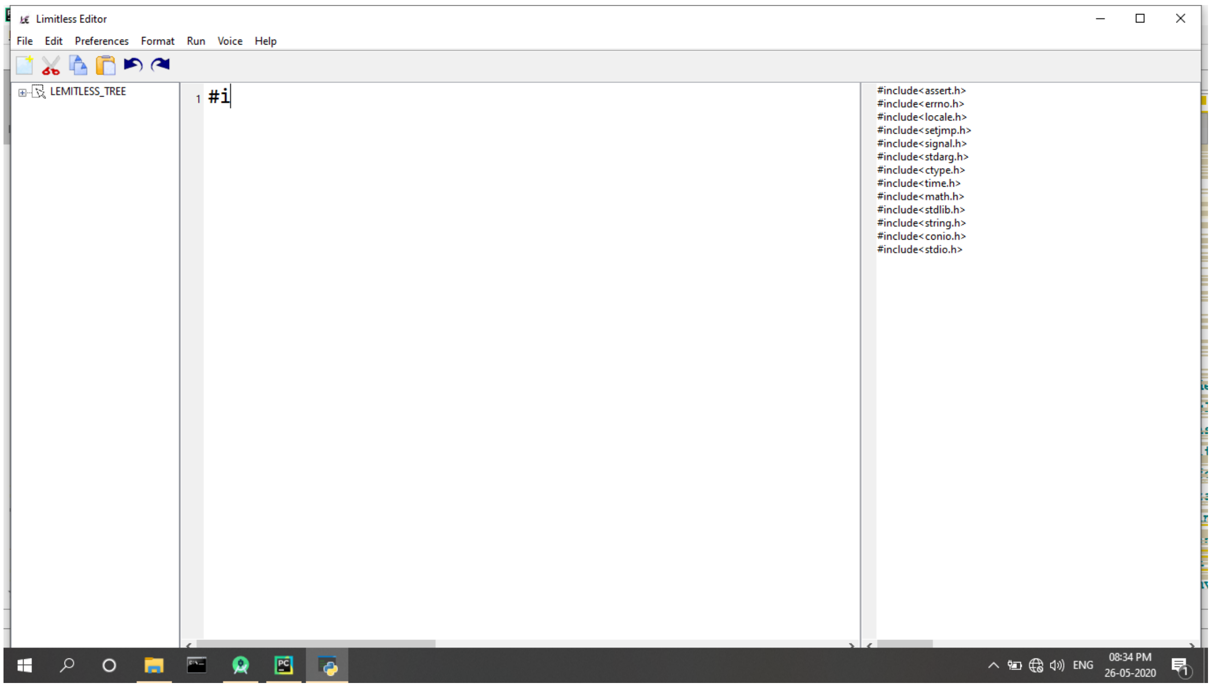


```
#include<stdio.h>
#include<conio.h>
void main()
{
printf("Hello world");
getch();
}
```

Name	Date modified	Type
BouncingBall.class	15-03-2020 12:14 PM	CLASS File
BouncingBall	15-03-2020 12:14 PM	JAVA File
Cal.class	24-03-2020 01:20 AM	CLASS File
Cal	13-04-2020 08:47 PM	JAVA File
Check.class	24-02-2020 11:13 PM	CLASS File
Check	24-02-2020 11:14 PM	JAVA File
Client	18-02-2020 12:27 PM	JAVA File
ClientData.class	11-03-2020 11:45 PM	CLASS File
ClientData	11-03-2020 11:43 PM	JAVA File
ClientData2.class	11-03-2020 11:46 PM	CLASS File

File name: temp.c
Save as type: All files (*.*)

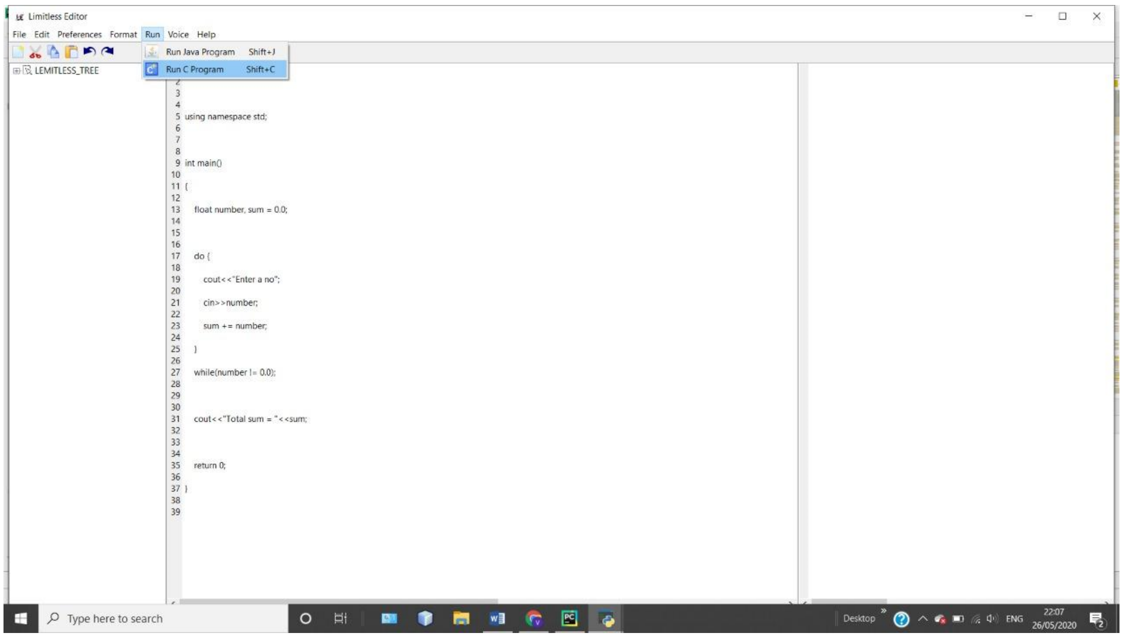
Figure 9.11: Save As Dialog



```
#i
```

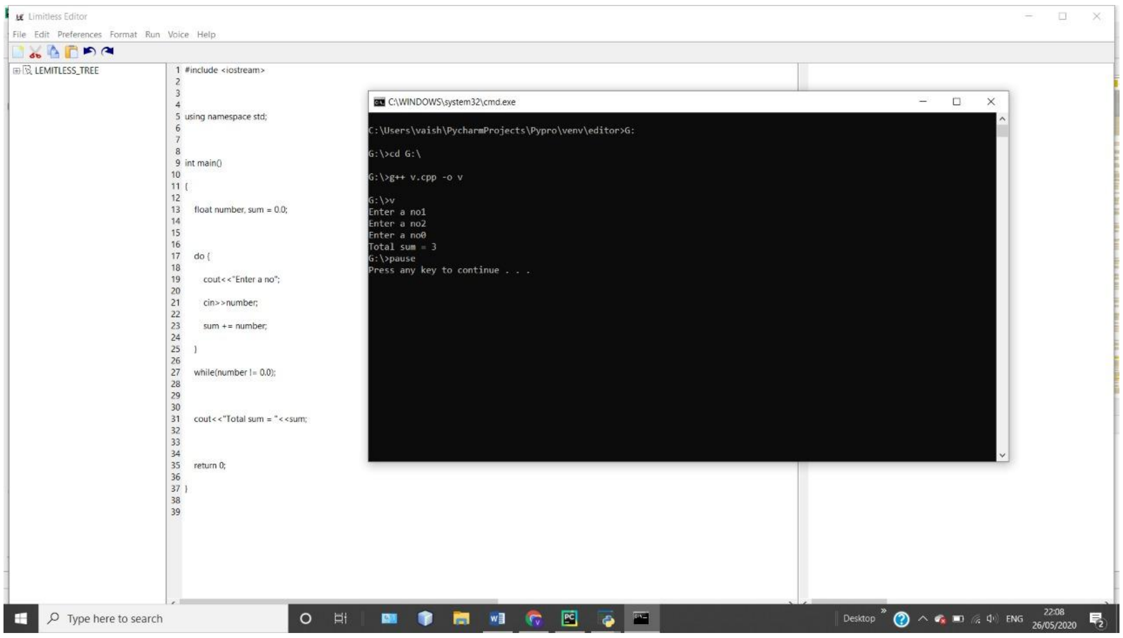
```
#include<assert.h>
#include<errno.h>
#include<locale.h>
#include<setjmp.h>
#include<signal.h>
#include<stdarg.h>
#include<ctype.h>
#include<time.h>
#include<math.h>
#include<stdlib.h>
#include<string.h>
#include<conio.h>
#include<stdio.h>
```

Figure 9.12: Suggestion Appearance



```
using namespace std;


int main()
{
   float number, sum = 0.0;

   do {
      cout<<"Enter a no";
      cin>>number;
      sum += number;
   }
   while(number != 0.0);

   cout<<"Total sum = "<<sum;

   return 0;
}
```

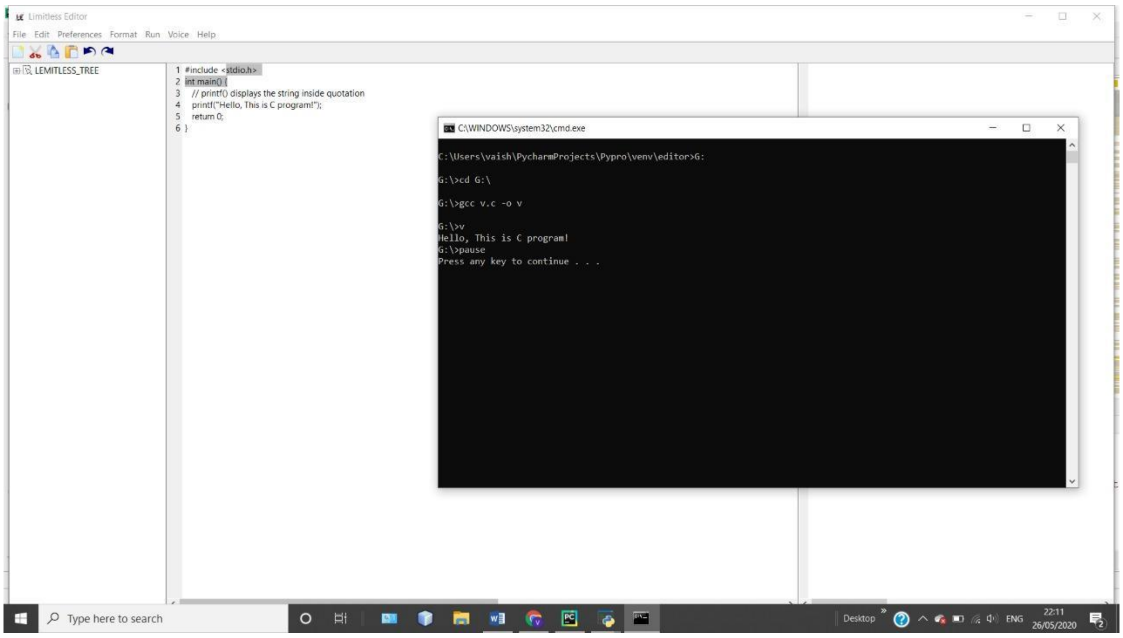
Figure 9.13: Run>Run C menu item


Limitless Editor
File Edit Preferences Format Run Voice Help
LEMITLESS_TREE
C:\WINDOWS\system32\cmd.exe

```
#include <iostream>

using namespace std;

int main()
{
   float number, sum = 0.0;

   do {
      cout<<"Enter a no";
      cin>>number;
      sum += number;
   }
   while(number != 0.0);

   cout<<"Total sum = "<<sum;

   return 0;
}
```

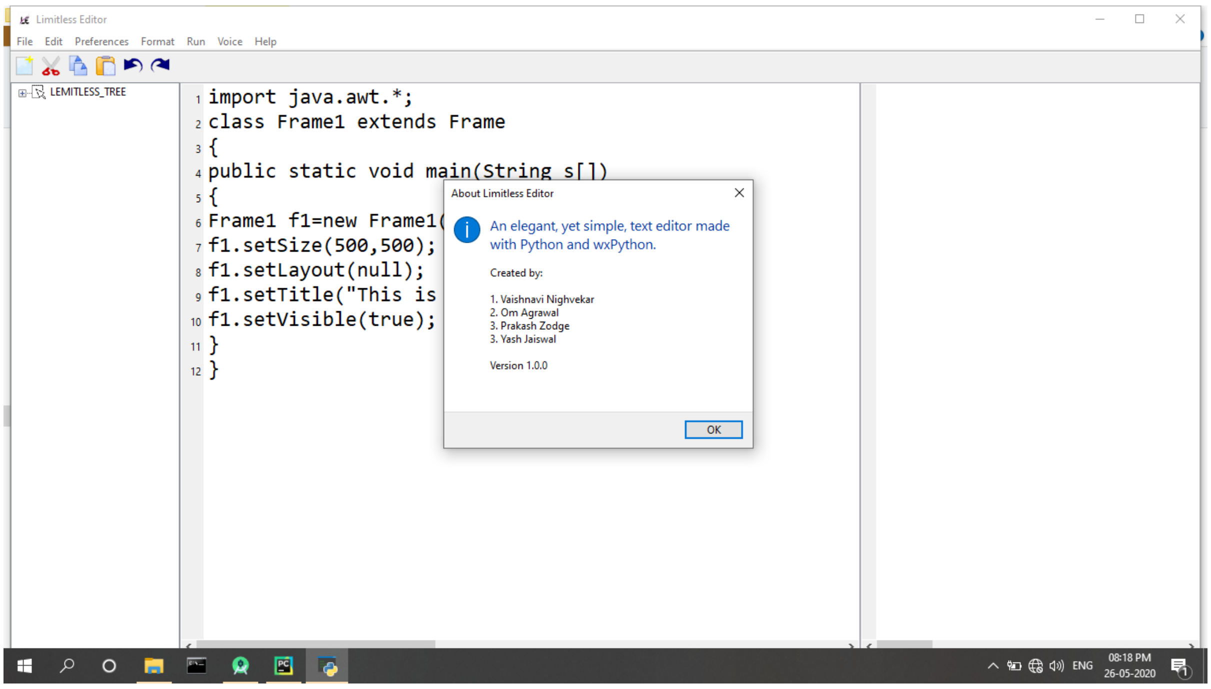
```
C:\Users\vaish\PycharmProjects\Pypro\venv\editor>G:

G:\>cd G:\

G:\>g++ v.cpp -o v

G:\>v
Enter a no1
Enter a no2
Enter a no0
Total sum = 3
G:\>pause
Press any key to continue . . .
```

Figuring 9.14: C++ Program Output


```c
#include <stdio.h>
int main() {
  // printf() displays the string inside quotation
  printf("Hello, This is C program!");
  return 0;
}
```

```
C:\Users\vaish\PycharmProjects\Pypro\venv\editor>G:

G:\>cd G:\

G:\>gcc v.c -o v

G:\>v
Hello, This is C program!
G:\>pause
Press any key to continue . . .
```

Figure 9.15: C Program Output


```java
import java.awt.*;
class Frame1 extends Frame
{
public static void main(String s[])
{
Frame1 f1=new Frame1(
f1.setSize(500,500);
f1.setLayout(null);
f1.setTitle("This is
f1.setVisible(true);
}
}
```

About Limitless Editor

An elegant, yet simple, text editor made with Python and wxPython.

Created by:

1. Vaishnavi Nighvekar
2. Om Agrawal
3. Prakash Zodge
3. Yash Jaiswal

Version 1.0.0

OK

Figure 9.16: About>About Limitless Editor

9.2 TESTING LIMITLESS EDITOR

9.2.1 WHAT IS THE TEST CASE?

A TEST CASE is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.

A test case can have the following elements.Note, however, that a test management tool is normally used by companies and the format is determined by the tool used.

Test Case ID	The ID of the test case.
Test Data	The test data, or links to the test data, that are to be used while conducting the test.
Expected Result	The expected result of the test.
Actual Result	The actual result of the test; to be filled after executing the test
TestCase Description	The summary / objective of the test case.
Status	Pass or Fail. Other statuses can be 'Not Executed' if testing is not performed and 'Blocked' if testing is blocked.

Table 9.2.1.1 : Test Case Description

9.2.2 TEST CASES FOR LIMITLESS EDITOR

The **Limitless Editor** is a type of very simple editor specially designed for java, C++ and C is used in console programing and normal cpp and java programs. Here we are going to test all the featuresand functions of the editor will help the user to type the program conveniently and efficiently.Test cases for operation of editor are as follows

Test Case for "New"

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Status
1	Check whether we can create a new file using "New" menu	File > New menu option	It should create a new blank file	It creates a new file	Pass
2	Check if user can open new file using keyboard shortcut	Press CTRL + N	It should open a new file	It creates a new file	Pass
3	Check whether it open new file by opening editor again	Open Limitless Editor	It should open a editor with new file	It opens a editor with new file	Pass

Table 9.2.2.1 : Test Case for "New"

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Status
1	Ensure shortcut key is worked to “Open” menu item	Press CTRL + O	Should open “Open” dialog box	Open dialog box executes	Pass
2	Ensure dialog box is closed when pressing Esc button in keyboard	Press Esc	Should close “Open” dialog box	Close “Open” dialog box	Pass
3	Check whether can we able to select multiple files at a movement	Try to select multiple file in “Open” dialog box	It should not allow to select multiple file	Does not allow to select multiple file	Pass

Table 9.2.2.2 : Test Case for “Open”

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Status
1	Check file is save after entering data	Open Limitless Editor >File > New > Enter Some Code and Click on “Save” Select path & give file name (“test.c”)	File should be saved in the specified location	File saved in the specified location	Pass
2	Check whether validation message has been generated on giving same name for saving	Open Limitless Editor File > Save > Save the file with existing file name (“test.c”)	Application should generate a validate message stating “Are you sure you want to replace the file” with yes/no option	Application generate a validate message stating “Are you sure you want to replace the file” with yes/no option	Pass
3	Verify that “Save” dialog box is opened when save option selection new file	1) Select file menu 2) Click on save menu item	It should open save dialog box	Opens save dialog box	Pass

Table 9.2.2.3 : Test Case for “Save”

Test Case for “Save As”

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Status
1	Check functionality of Save as	1) File 2) Open item 3) Select Save as 4) Select path & give file name 5) Click on save button	File should be saved in the specified location	File saved in specified location	Pass
2	Check for rename the file using Save as	1) Open existing file 2) Click on “Save As” 3) Edit the file name	It should save file with new name i.e. updated name	Save the file with updated name	Pass
3	Check whether it allow the user to change file format	And save the “.C” file with “.java” extension	It should allow to change the file format	Allow to change the file format	Pass

Table 9.2.2.4 : Test Case for “Save As”

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Status
1	Check that the application is closed by clicking on “Exit” menu	1) Select File 2) Click on “Exit”	Should close Limitless Editor	Close Limitless Editor	Pass
2	Verify the application support shortcut key to close the Limitless Editor	Press CTRL + X	Should close Limitless Editor	Close Limitless Editor	Pass
3	Check whether it support to Exit application by using ALT + F4	Press ALT + F4	Should close Limitless Editor	close Limitless Editor	Pass

Table 9.2.2.5 : Test Case for “Exit”

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Status
1	Verify that user can select text then cut by using shortcut key	Press CTRL + X to cut the selected text	Should cut the selected text	It cut the selected text	Pass
2	Verify that user can paste the copied text multiple time	1) Select the text 2) Select "Copy" 3) Select "Paste" option	Should paste the copied text	Paste the copied text	Pass
3	Check whether it allow to copy text using shortcut key	1) Select the text 2) Press Ctrl + X 1)	Should copy text	It copies the selected text	Pass
4	Check "Copy" option in edit menu	1) Select the text 1) Click on "Copy "	It should copy the text without deleting it	It copies the selected text	Pass
5	Use CTRL + V hotkey without selecting & copying text	Press CTRL + V without copying any text	It should not paste anything	Didn't paste anything	Pass

Table 9.2.2.6 : Test Case for "Cut/Copy & Paste"

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Status
1	Checking bat file of C Program was run properly	1) Enter Code of C 2) Click "Run C Program"	Execute "c.bat" file	Executed "c.bat" file successfully	Pass
2	Checking bat file of Java Program was run properly	1) Enter Code of Java 2) Click "Run Java Program"	Execute "java.bat" file	Executed "java.bat" file successfully	Pass
3	Verify short cut key of "Run C Program" menu item	Press SHIFT +C	Execute "c.bat" file	Executed "c.bat" file successfully	Pass
4	Verify short cut key of "Run Java Program" menu item	Press SHIFT + J	Execute "java.bat" file	Executed "java.bat" file successfully	Pass

Table 9.2.2.7 : Test Case for "Run"

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Status
1	Checking Visibility of tree	Click on “+” of tree	Displays all tree	Display tree items	Pass
2	Check whether it will display the selected text on plain text area using mouse	1) Click on “+” of tree 2) Double-click on any text	Displays selected text on Plane text area	Showing selected text on plain text area	Pass
3	Check whether it will display the selected text on plain text area using keyboard key	1) Click on “+” of tree 2) Press Enter on any text	Displays selected text on Plane text area	Showing selected text on plain text area	Pass
4	Checking single click on tree	1) Click on “+” of tree 2) Single-click on any text	It should not display the tree on single click	Not displayed	Pass

Table 9.2.2.8 : Test Case for “Tree”

Test Case No.	Test Case Description	Test Data	Expected Result	Actual Result	Status
1	Check whether it convert the text in male voice.	Click on Voice menu > Male menu item	It should speak out the text in male voice	Speak out the text in male voice	Pass
2	Check whether it convert the text in female voice	Click on Voice menu> Female menu item	It should speak out the text in female voice	Speak out the text in female voice	Pass
3	Checking shortcut key for Male Voice	Press CTRL + SHIFT + M	It should speak out the text in male voice	Speak out the text in male voice	Pass
4	Checking shortcut key for Female Voice	Press CTRL + SHIFT + F	It should speak out the text in female voice	Speak out the text in female voice	Pass

Table 9.2.2.9 : Test Case for “Voice”

CHAPTER 10
CONCLUSION

- From this project we tried to recover some drawbacks of existing editors and make the editor lightweight.
- In future we will try to add multiple languages and features in our editor to make it more perfect.
- In this project we have also learned many new concepts of python and successfully developed The Limitless editor with the help of python. This project will be a useful for all the programmers, Its various functions like voice speech, suggestions, tree copy will help the programmers to write their code more easily and more faster than any other editors.
- In future it the demand of the limitless editor increases then our group will try to to include other funcitionality such as voice typing and error line guesing and many other features, after that we will also try to launch the limitless editor on any big platform so that this editor can also help many other programmers for their programming world expirence.

CHAPTER 11
BIBLIOGRAPHY

- **BOOKS**

1. Learning with Python 3 - Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers
2. Python 3 – Object Oriented Programming Third Edition – Dusty Phillips

- **WEB REFERENCES**

1. https://www.geeksforgeeks.org/file-handling-python/
2. https://wxpython.org/Phoenix/docs/html/wx.stc.StyledTextCtrl.html
3. https://wxpython.org/Phoenix/docs/html/wx.1moduleindex.html
4. https://realpython.com/python-gui-with-wxpython/