Project Title :- NOTEPAD

Faculty : AMARINDER KAUR Ma'am

<u>Group Member</u>

Name	Register Number	Roll no
1. Vikash pandey	12106224	RK21ANB67
2. KRISNANDAN SAH KANU	12106329	RK21ANB46
3. MUKUL	12113606	RK21ANB39

INTRODUCTION :-

• Notepad is a simple text editor application used to create and edit plain text files.

• It was created by Microsoft and has been a part of the Windows operating system since Windows 1.0 in 1985.

• In this presentation, we will be discussing the implementation of Notepad using Java language.

Why we choose java for NOTEPAD :-

❑ **As we all know that Java is a popular programming language that has several features that make it well-suited for steganography applications.**

1. Java is platform-independent, meaning that the code written in Java can be executed on any platform without the need for recompilation.

2. Java has a rich set of libraries and APIs that provide developers with a wide range of functionalities to work with.

3. Java has strong support for object-oriented programming, which allows developers to create modular and reusable code that can be easily extended and maintained over time.

4. Java has a strong security model that makes it well-suited for notepad applications.

Advantages of NOTEPAD :-

1.Notepad is a reasonably simple tool, making it an excellent for programmer / students who are just getting started with coding.

2.Customizability: Even though notepad is a simple program, it may be tailored to the needs of the user. Developers can enhance the application in future.

3.Accessibility: Because building a notepad project can make the application more accessible to people who may not have access to more specialized applications.

So a notepad project can be a useful tool for both developers and users, providing a simple, customizable, and accessible text editor that can be used for a variety of tasks.

Creating NOTEPAD using Java :-

•To create Notepad using Java, we will use the Swing toolkit, which is a set of graphical user interface (GUI) components for Java.

•We will create a J-Frame object to represent the main window of the Notepad application.

•We will add a J-TextArea object to the J-Frame to allow users to enter and edit text.

•We will also add a J-MenuBar object to the J-Frame to provide users with access to various file operations, such as opening and saving files.

MODULES :-

Module 1: **User Interface**
Create a graphical user interface (GUI) using Swing or JavaFX.Include a menu bar with options such as File, Edit, and Help.Add menu items such as New, Open, Save, Save As, Cut, Copy, Paste, and Exit.Add a text area to the GUI where the user can type and edit text .
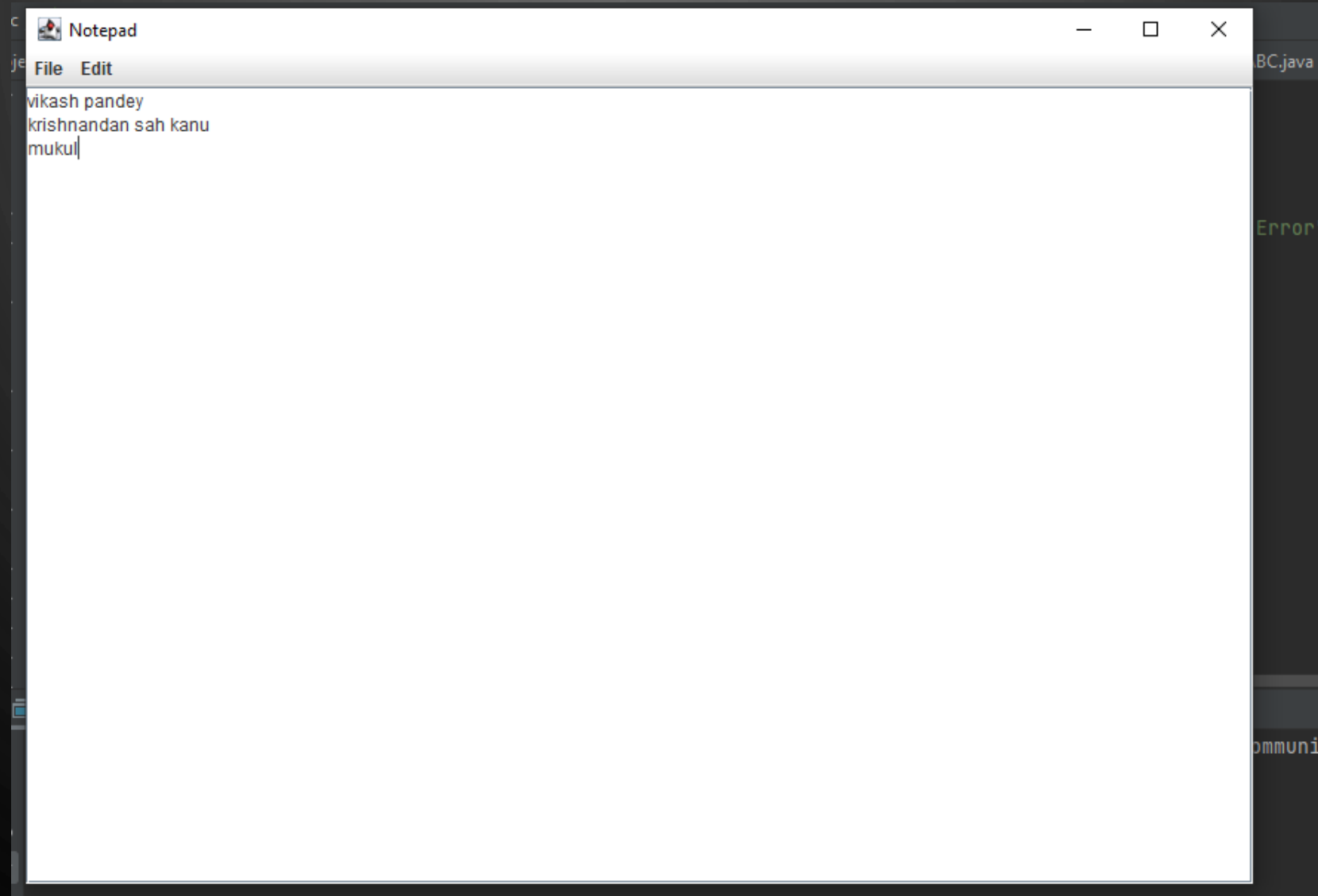
Module 2: **File Operations**
Implement the functionality for opening and saving files.When the user selects the Open menu item, display a file chooser dialog box that allows them to select a file to open.When the user selects the Save or Save As menu item, display a file chooser dialog box that allows them to specify a filename and location to save the file.Use FileReader and FileWriter classes to read and write files .

Module 3: **Edit Operations**
Implement the functionality for cut, copy, and paste operations.Use the JTextArea's built-in cut(), copy(), and paste() methods to perform these operations.Add a context menu that appears when the user right-clicks on the text area, which includes options to cut, copy, and paste.
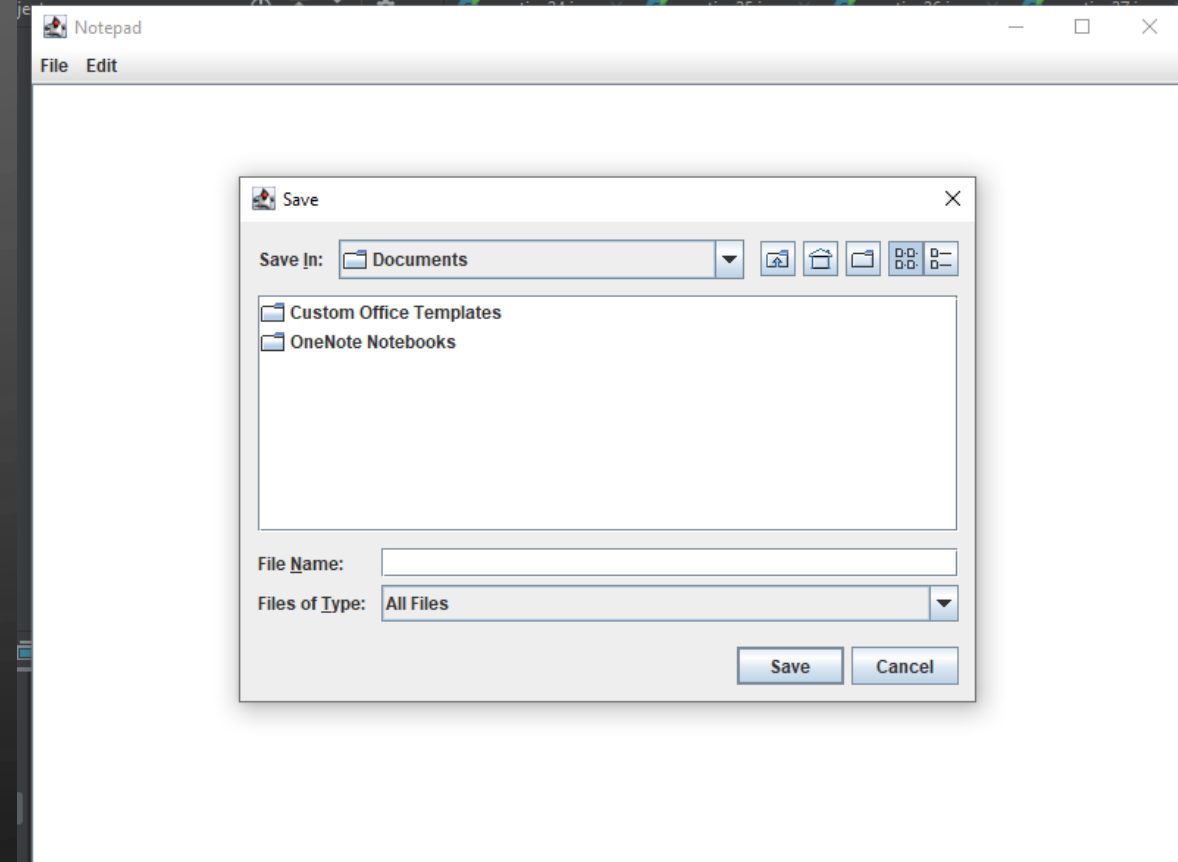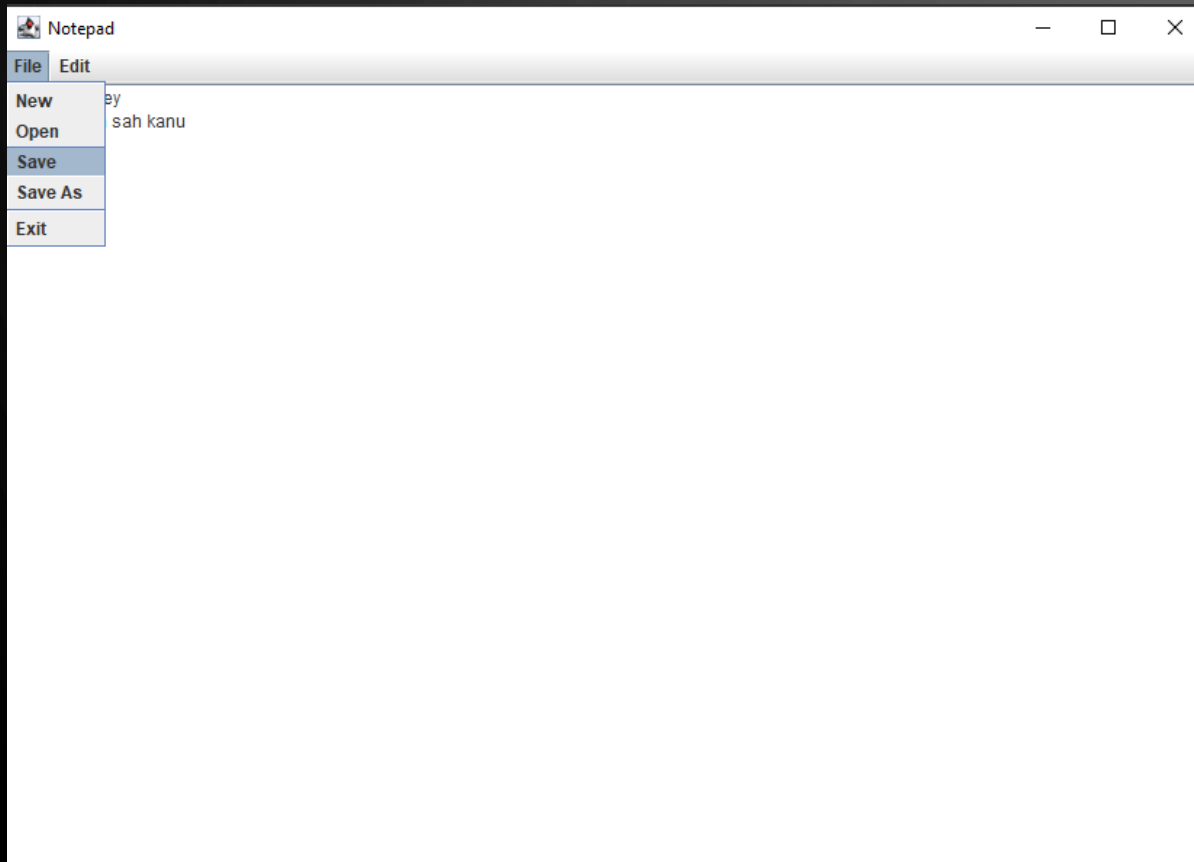
RESULTS (SCREENSHOTS) :-
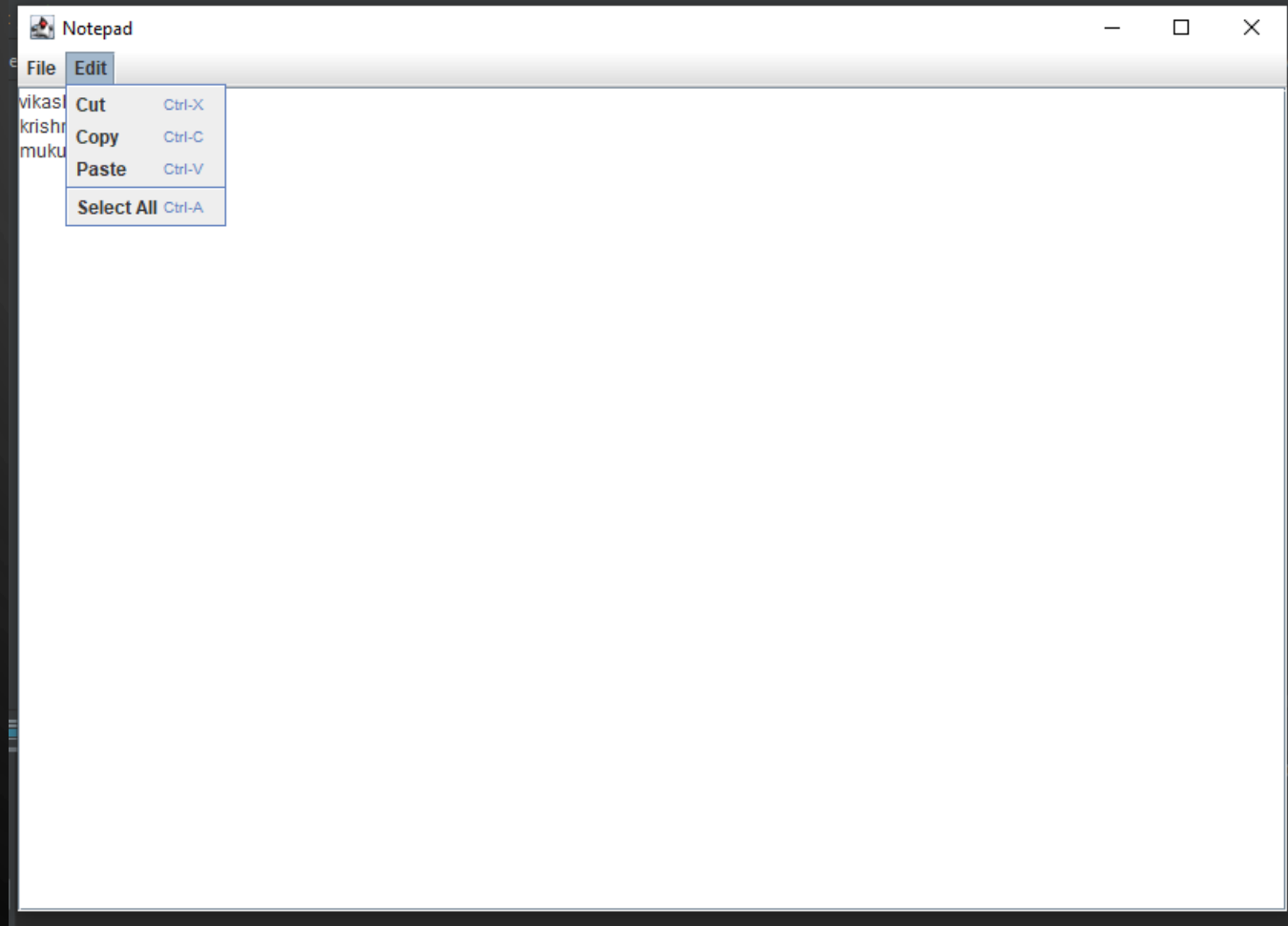## MAIN SCREEN(MODULE 1)

RESULTS (SCREENSHOTS) :-
MAIN SCREEN(MODULE 2)

RESULTS (SCREENSHOTS) :-
MAIN SCREEN(MODULE 3)

CONCLUSION :-

• In this presentation, we have discussed the implementation of Notepad using Java language.

• We have seen how to use the Swing toolkit to create a graphical user interface for the Notepad application.

• We have also seen a code example that demonstrates how to create a basic Notepad application using Java.

• With this knowledge, you can create your own Notepad application or customize the code to suit your needs.

REFRENCES :-

1. GUI Refrence from Youtube .
2. www.W3school.com
3. www.Javatpoint.com