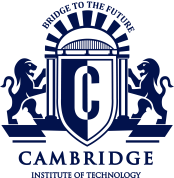
**Cambridge Institute of Technology**

**Department of Computer Science and Engineering**

****

**Project Report**

**on**

**TIC TAC TOE**

by

**Namita Annu Shaji (1CD23CS101)**

**Neelakshi Yemde (1CD23CS104)**

**Prabin Dahal(1CD23CS118)**

**Section:3B**

**Under the Guidance of**

**Prof. Arun P**

**- Asst. Professor**

Dept. of CSE

AIM:-

This project aims to create an account in GitHub and perform some basic operations of file management using the command prompt of the respective Operating Systems.

DESCRIPTION:-

* In this project we are going to use the Git-Bash prompt to Create a repository in GitHub and insert some files and folders into it manually .
* We then add the folders and files into the repositories using the Git-Bash commands such as init, mkdir, add., etc…

METHOD:-

1. INSTALLING GIT:

To Install the GIT on your system go through the following procedures:

1. Go to this website https://git-scm.com/downloads .
2. Select the files as per the Operating System you are using e.g. Windows, macOS, Linux.
3. Now click on the Downloaded file and install the GIT software.
4. Click on the yes button on the pop-up window.
5. Click on accept.
6. Click on the next button.
7. TO CREATE AN ACCOUNT IN GITHUB:

To create an account in the GITHUB go through the following procedures:

1. Go to https://github.com/.
2. Click Sign up.
3. Enter your email address, a username, and a password.
4. Select Continue.
5. Verify your account by solving a puzzle.
6. GitHub will send a launch code to your email address.
7. Enter the Launch code in the GitHub prompt.
8. GitHub will ask you some questions to tailor experience.
9. Choose whether you want to use the Free Account or the Team account.
10. Now sign-in the account with the email address and the password that you used in creating your GitHub account.

• BASIC COMMANDS:-

1)Initialize a GIT Repository:

git init: Initializes a new Git repository in the current directory.

2) Create Files and Folders:

a) touch filename: Create a file.

b) mkdir FolderName: Creates a folder.

c) touch foldername/filename : Create a file inside the folder.

3) Stage Files:

git add . : Stages all the files and folders for committing.

4) Commit Changers:

git commit –m “Commit message”: Saves the staged changes with description.

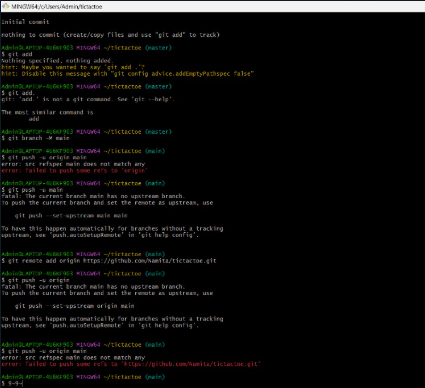
5) Connect To GitHub Repository:

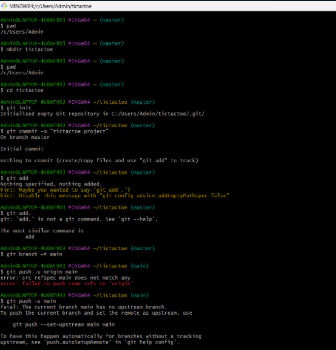
git remote add origin : links the URL of local repository.

6) Push Changes:

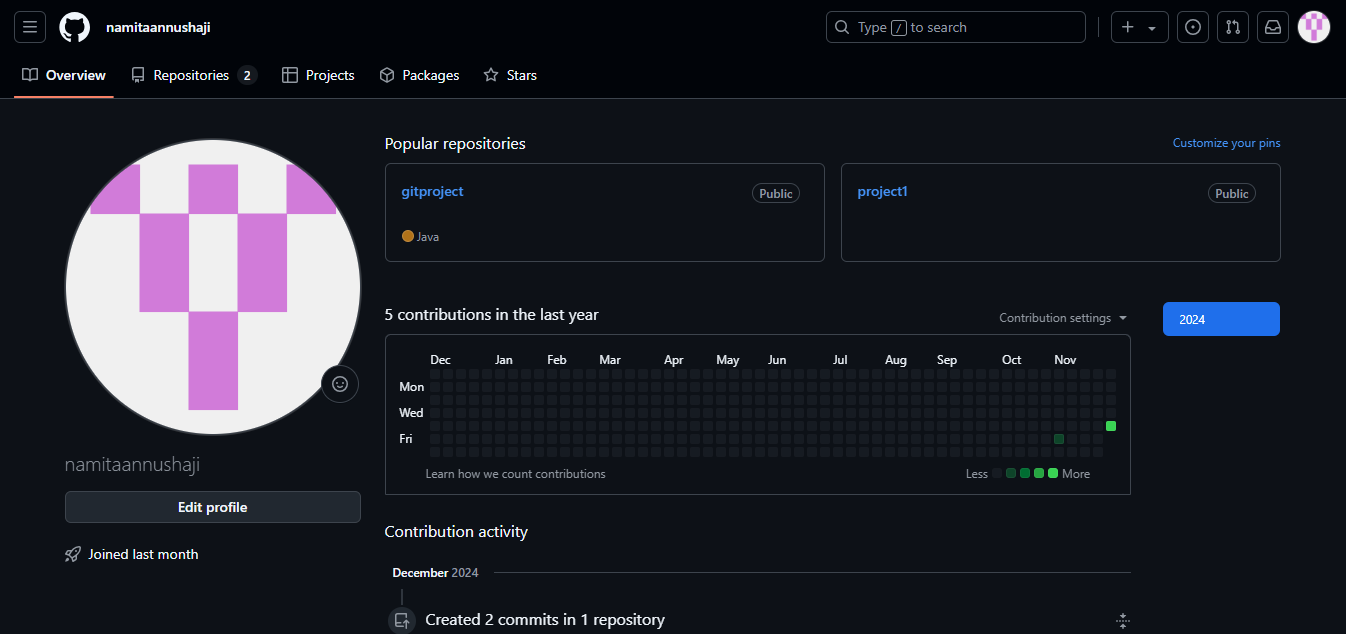
git branch –M main: Renames the branch to main.

git push –u origin main: Pushes the committed changes to the main branch on GitHub.

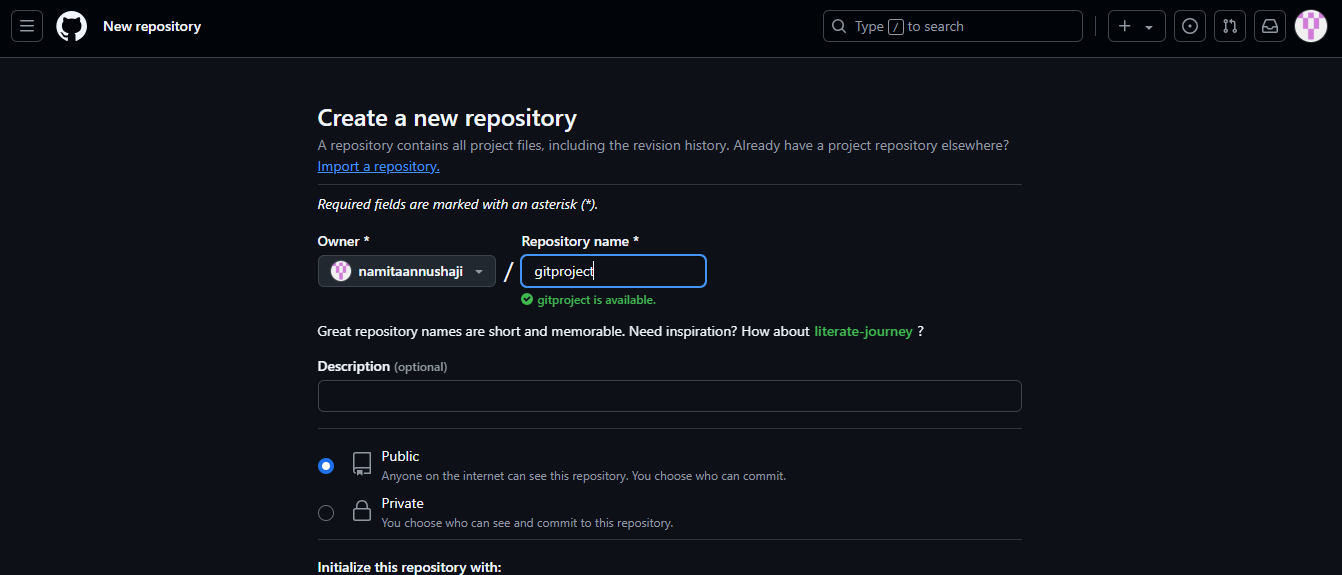




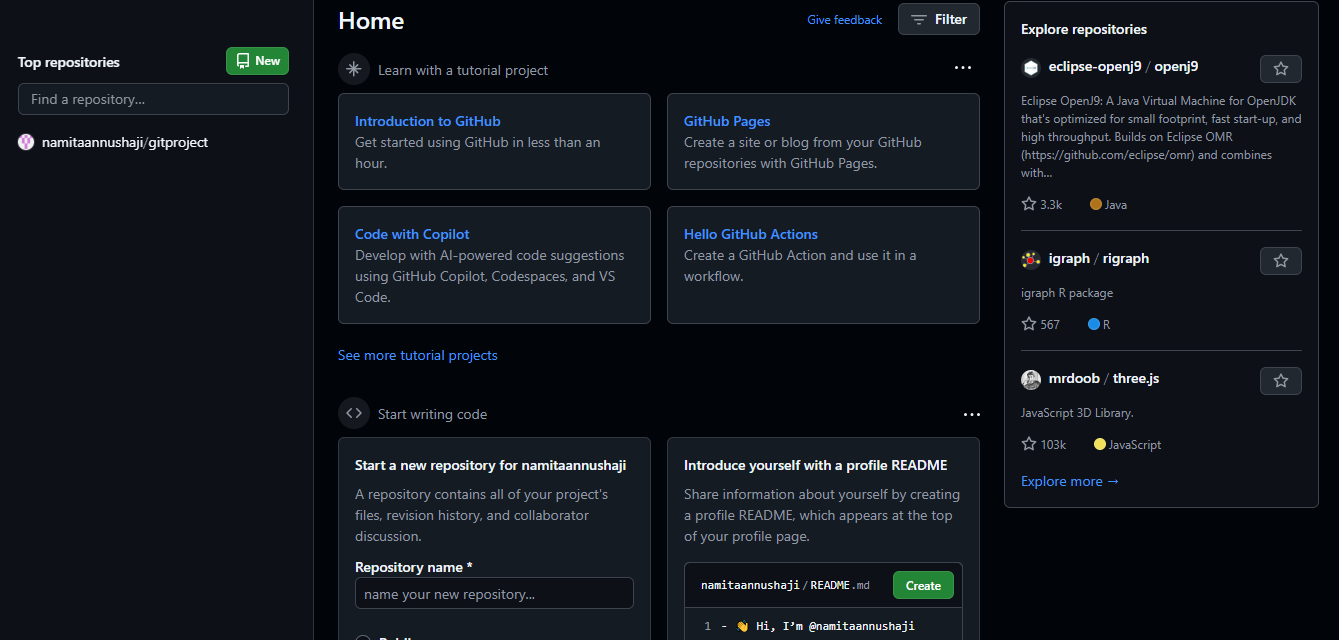
**PROFILE:**

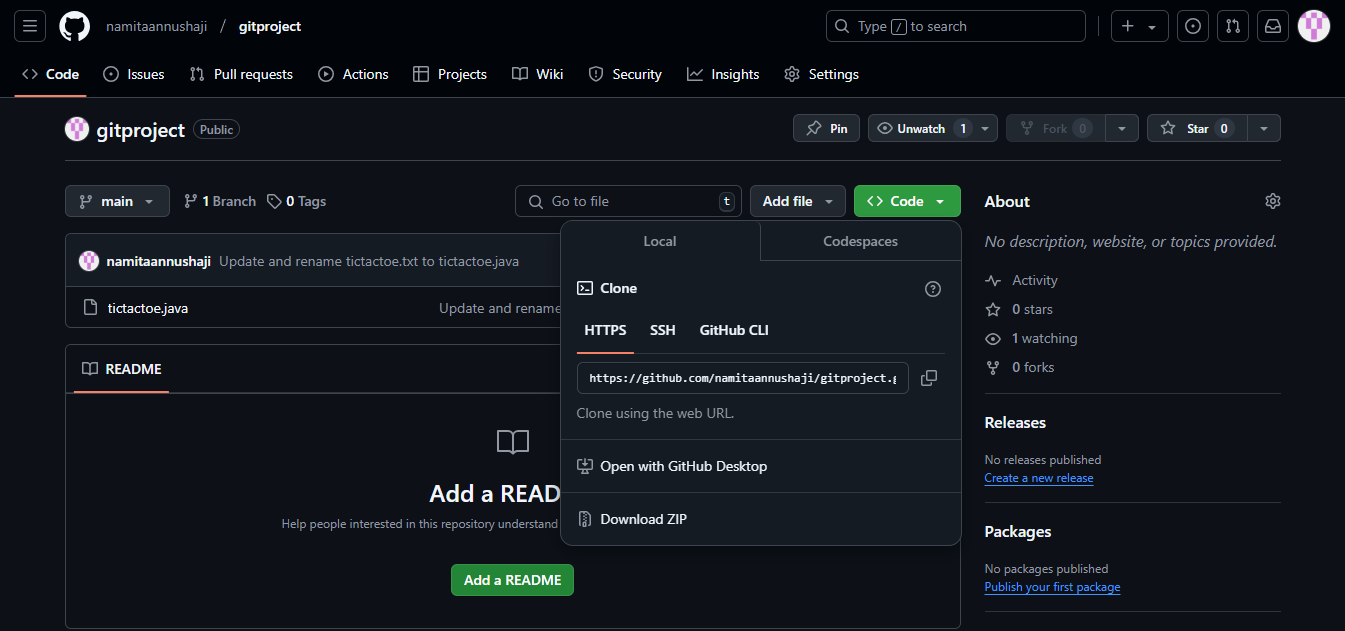


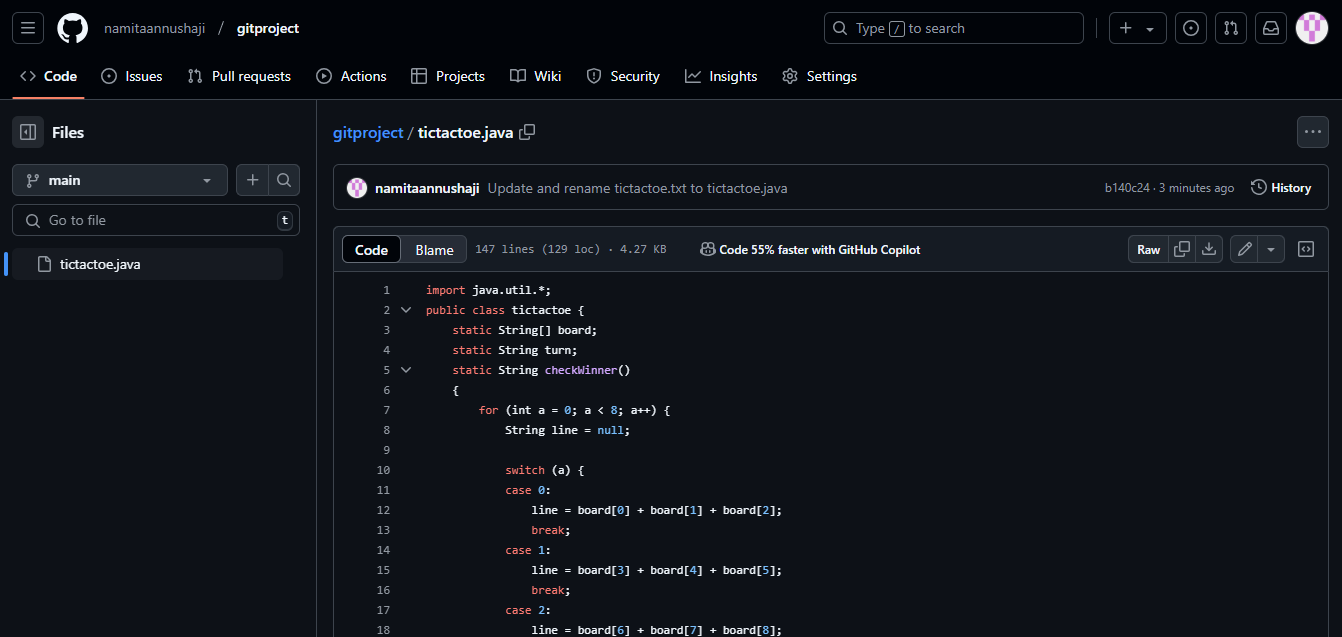
**NEW REPOSITORY:**



**GIT INTERFACE:**







CONCLUSION: We have successfully created a account in GitHub, created a repository, added files into it, pushed folders into it.

**DESCRIPTION:**

**Tic Tac Toe** is a simple two-player game played on a 3x3 grid. The players take turns marking a square on the grid, with one player using "X" and the other using "O."

**METHOD:**

1. SETTING UP THE JAVA DEVELOPMENT ENVIRONMENT:

To begin developing the Hangman game in Java, make sure you have Java installed on your system. If not, follow these steps to set it up:

INSTALLING JAVA:

• Go to the official Java download page: https://www.oracle.com/java/technologies/javase-jdk11- downloads.html

• Download the appropriate version of Java for your operating system (Windows, macOS, or Linux).

• Install Java by following the installation wizard.

• To verify that Java is installed correctly, open the command prompt (Windows) Java is installed and ready for use.

**Rules:**

 The grid starts empty.

 Players decide who will use X and who will use O.

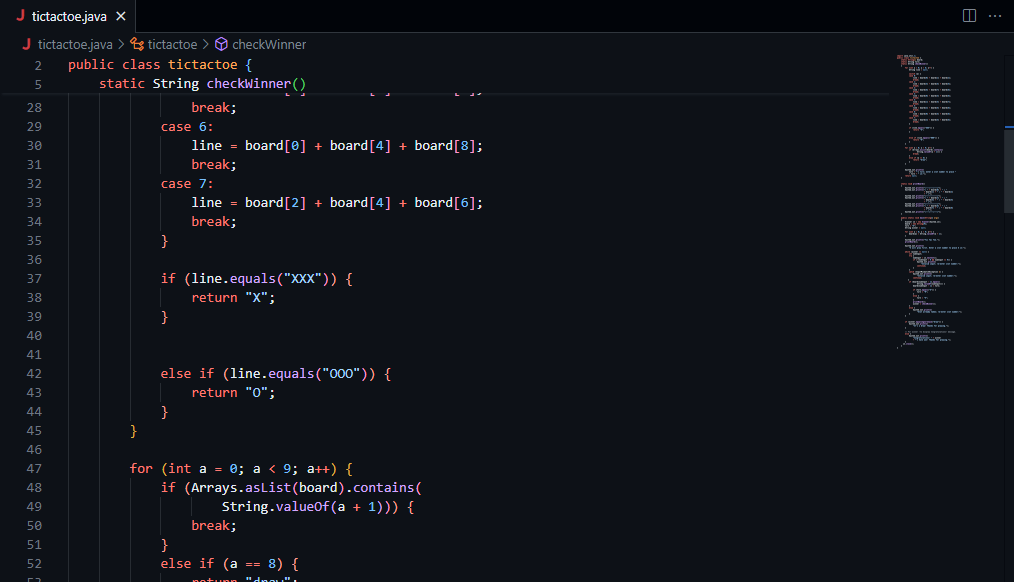
 Players alternate turns, placing their mark in an empty square.

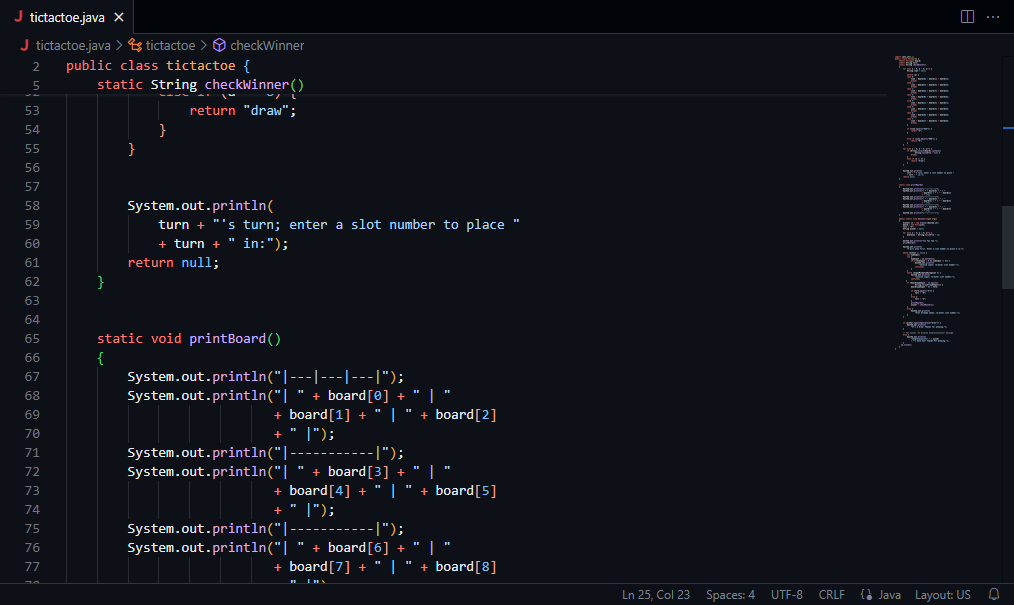
 The game ends when:

* A player forms a line of three marks in a row, column, or diagonal (they win), or
* All squares are filled without a winner (the game is a tie).

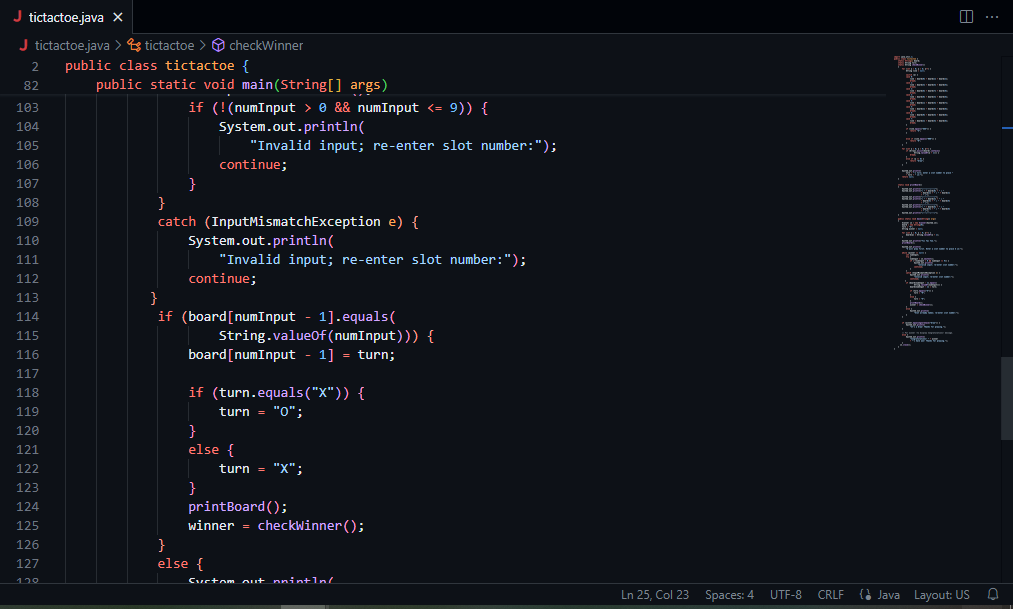
**PROGRAM**

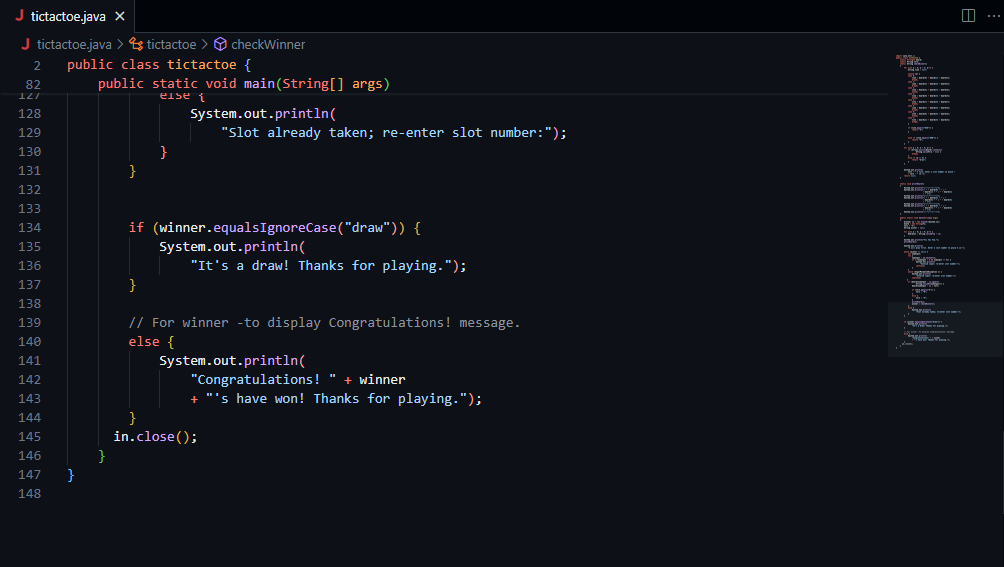












**Output**

