Sudoku Solver - CODECRAFT_TASK_04

TASK - 04: Implement a Sudoku Solver with GUI

This project implements a Sudoku Solver using the Backtracking algorithm.

The program features a graphical user interface (GUI) built with tkinter,

allowing users to input Sudoku puzzles and solve them with a click of a button.

Algorithm Explanation:

The Backtracking algorithm is a brute-force recursive method to solve constraint satisfaction problems.

In Sudoku, the algorithm works as follows:

1. Find the first empty cell.

2. Try filling it with digits 1 through 9.

3. Check if the digit is valid (row, column, and 3x3 box constraints).

4. If valid, recurse to solve the next cell.

5. If not solvable, backtrack and try another digit.

6. Continue until the board is completely filled.

Python Code:

```
import tkinter as tk
from tkinter import messagebox

# Backtracking Sudoku Solver
def is_valid(board, row, col, num):
    for i in range(9):
        if board[row][i] == num or board[i][col] == num:
            return False

    start_row = row - row % 3
    start_col = col - col % 3

    for i in range(3):
        for j in range(3):
            if board[start_row + i][start_col + j] == num:
                return False

    return True

def solve_sudoku(board):
    for row in range(9):
```

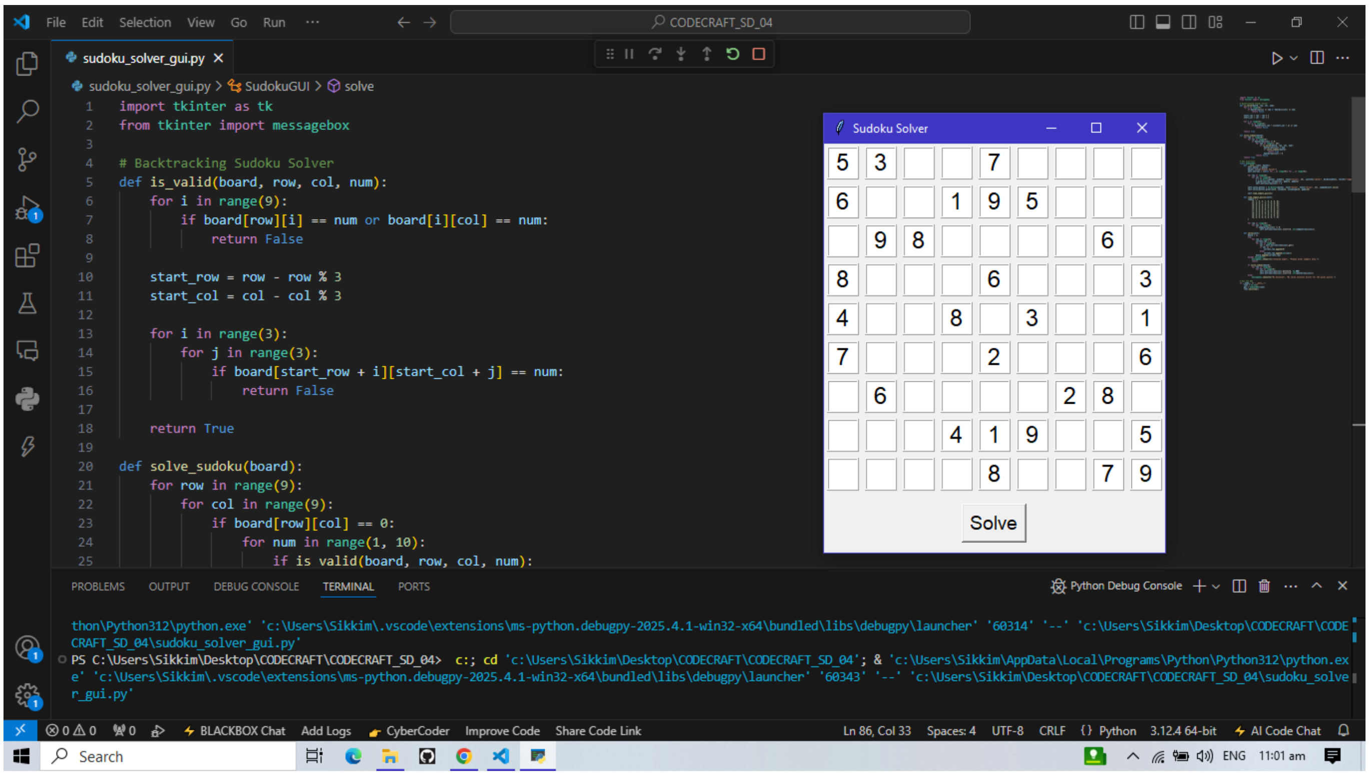
Sudoku Solver - CODECRAFT_TASK_04

```
        for col in range(9):
            if board[row][col] == 0:
                for num in range(1, 10):
                    if is_valid(board, row, col, num):
                        board[row][col] = num
                        if solve_sudoku(board):
                            return True
                        board[row][col] = 0
                return False
    return True

# GUI Interface
class SudokuGUI:
    def __init__(self, master):
        self.master = master
        master.title("Sudoku Solver")
        self.entries = [[None for _ in range(9)] for _ in range(9)]

        for row in range(9):
            for col in range(9):
                e = tk.Entry(master, width=2, font=("Arial", 18), justify="center")
                e.grid(row=row, column=col, padx=5, pady=5)
                self.entries[row][col] = e

        self.solve_button = tk.Button(master, text="Solve", command=self.solve)
        self.solve_button.grid(row=9, column=0, columnspan=9, pady=10)

    def solve(self):
        board = []
        try:
            for row in range(9):
                current_row = []
                for col in range(9):
                    val = self.entries[row][col].get()
                    if val == "":
                        current_row.append(0)
                    else:
                        current_row.append(int(val))
                board.append(current_row)
        except ValueError:
            messagebox.showerror("Invalid Input", "Please enter numbers only.")
            return

        if solve_sudoku(board):
            for row in range(9):
                for col in range(9):
                    self.entries[row][col].delete(0, tk.END)
                    self.entries[row][col].insert(0, str(board[row][col]))
        else:
            messagebox.showinfo("No Solution", "No valid solution exists for the given puzzle.")
```

```
# Run the GUI
if __name__ == "__main__":
    root = tk.Tk()
    gui = SudokuGUI(root)
    root.mainloop()
```

Sudoku Solver - CODECRAFT_TASK_04

Screenshots:



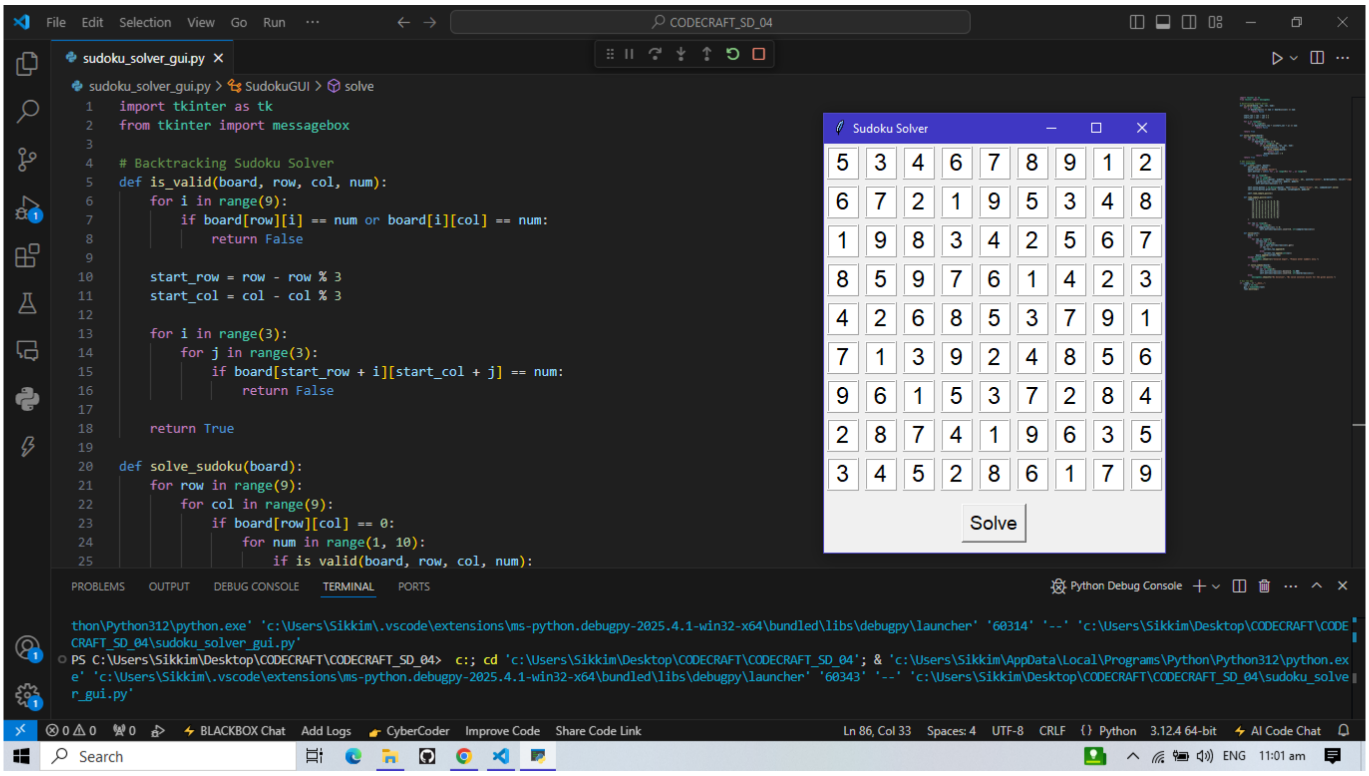
```python
import tkinter as tk
from tkinter import messagebox

# Backtracking Sudoku Solver
def is_valid(board, row, col, num):
    for i in range(9):
        if board[row][i] == num or board[i][col] == num:
            return False

    start_row = row - row % 3
    start_col = col - col % 3

    for i in range(3):
        for j in range(3):
            if board[start_row + i][start_col + j] == num:
                return False

    return True

def solve_sudoku(board):
    for row in range(9):
        for col in range(9):
            if board[row][col] == 0:
                for num in range(1, 10):
                    if is_valid(board, row, col, num):
```

Sudoku Solver

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

```
thon\Python312\python.exe' 'c:\Users\Sikkim\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundled\libs\debugpy\launcher' '60314' '--' 'c:\Users\Sikkim\Desktop\CODECRAFT\CODECRAFT_SD_04\sudoku_solver_gui.py'
PS C:\Users\Sikkim\Desktop\CODECRAFT\CODECRAFT_SD_04>  c:; cd 'c:\Users\Sikkim\Desktop\CODECRAFT\CODECRAFT_SD_04'; & 'c:\Users\Sikkim\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\Sikkim\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundled\libs\debugpy\launcher' '60343' '--' 'c:\Users\Sikkim\Desktop\CODECRAFT\CODECRAFT_SD_04\sudoku_solver_gui.py'
```

File Edit Selection View Go Run
CODECRAFT_SD_04
sudoku_solver_gui.py
sudoku_solver_gui.py > SudokuGUI > solve
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link
Ln 86, Col 33 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit AI Code Chat
Search
ENG 11:01 am



```python
import tkinter as tk
from tkinter import messagebox

# Backtracking Sudoku Solver
def is_valid(board, row, col, num):
    for i in range(9):
        if board[row][i] == num or board[i][col] == num:
            return False

    start_row = row - row % 3
    start_col = col - col % 3

    for i in range(3):
        for j in range(3):
            if board[start_row + i][start_col + j] == num:
                return False

    return True

def solve_sudoku(board):
    for row in range(9):
        for col in range(9):
            if board[row][col] == 0:
                for num in range(1, 10):
                    if is_valid(board, row, col, num):
```

Sudoku Solver

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

```
thon\Python312\python.exe' 'c:\Users\Sikkim\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundled\libs\debugpy\launcher' '60314' '--' 'c:\Users\Sikkim\Desktop\CODECRAFT\CODECRAFT_SD_04\sudoku_solver_gui.py'
PS C:\Users\Sikkim\Desktop\CODECRAFT\CODECRAFT_SD_04>  c:; cd 'c:\Users\Sikkim\Desktop\CODECRAFT\CODECRAFT_SD_04'; & 'c:\Users\Sikkim\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\Sikkim\.vscode\extensions\ms-python.debugpy-2025.4.1-win32-x64\bundled\libs\debugpy\launcher' '60343' '--' 'c:\Users\Sikkim\Desktop\CODECRAFT\CODECRAFT_SD_04\sudoku_solver_gui.py'
```

File Edit Selection View Go Run
CODECRAFT_SD_04
sudoku_solver_gui.py
sudoku_solver_gui.py > SudokuGUI > solve
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python Debug Console
BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link
Ln 86, Col 33 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit AI Code Chat
Search
ENG 11:01 am