1.Project Description

This project is an interactive tool designed to solve the map coloring problem using advanced algorithms such as the Four Color Theorem and Constraint satisfaction problem Algorithm (CSP) . Regions or nodes are represented as a graph, and different colors are assigned to each region such that no two adjacent regions share the same color. The project provides an easy-to-use graphical user interface (GUI) that allows users to input data about regions and their borders, and automatically generates the coloring.

2. Tools and Libraries Used

Python: Primary programming language.

Tkinter: For creating the graphical user interface (GUI).

NetworkX: For graph representation and data processing.

Matplotlib: For visualizing graphs and maps within the GUI.

Messagebox: For displaying alerts and notifications.

FigureCanvasTkAgg: To integrate Matplotlib plots into the Tkinter interface.

➢ **PEAS** (Performance, Environment, Actuators, Sensors)

Component	Description
Performance	- Safe and accurate coloring. - Efficient use of minimum colors. - Intuitive GUI. - Fast processing.
Environment	- Regions and borders entered by the user. - Graph structure representing regions and adjacency. - Predefined color palette.
Actuators	- Buttons for adding regions, borders, and generating the map. - Coloring algorithm to assign colors. - Real-time visualization.
Sensors	- Text fields for region and border input. - Button clicks detected by the system. - Graph data processed from user inputs.

➢ **ODESDA** (Observable, Deterministic, Episodic, Static, Discrete, Agent)

Component	Description
O	**Fully observable**
D	**Deterministic**
E	**Sequential**
S	**Static**
D	**Discrete**
A	**Single**

➢ **Agent Type:** Goal-Based Agents