The project title: Map Coloring Puzzle Solver

The URL to the GitHub repository:

https://github.com/ankith-i/CS-161-Sec-01-Ankith-Indrakumar

Milestones with deadlines:

M1 (Setup and Initial Deployment): Setup basic Flask app structure, initial deployment to Heroku. Deadline: Week 1

M2 (Game Logic Implementation): Implement the map coloring logic and a simple user interface. Deadline: Week 2

M3 (User Accounts and Persistence): Add functionalities for user registration, login, and saving game states. Deadline: Week 3

M4 (Solver and Hints): Integrate the backtracking algorithm for solving puzzles and providing hints. Deadline: Week 4

M5 (Final Touches and Deployment): Final testing, UI/UX enhancements, and deployment. Deadline: Week 5

Algorithms/AI schemes used in the core engine:

1. Utilizes a backtracking algorithm for solving the map coloring puzzle, suitable for demonstrating constraint satisfaction problem-solving.

Describe the market space the application is related to and the selling points of your product:

1. The application is aimed at the educational market, specifically targeting those interested in puzzles, computer science, and mathematics.
2. Selling points include its educational value, engagement through interactive problem-solving, and insights into algorithmic thinking.

Tech Stack

Frontend Technologies:

1. React: Utilized for building the user interface, offering a dynamic and responsive web experience. React's component-based architecture facilitates efficient updates and rendering of the interactive elements of the map coloring puzzle.
2. HTML/CSS: Integrated within React components for structuring and styling, leveraging modern CSS frameworks or custom styles for a polished look.

3. JavaScript (ES6+): For adding interactivity, managing state, and handling events within the React application.
4. Bootstrap or Tailwind CSS (optional): For rapid development of responsive and aesthetically pleasing user interfaces.

Backend Technologies:

1. Flask: Serves as the backend framework, handling HTTP requests, API endpoints for game logic, user authentication, and interaction with the database.
2. Python: The primary programming language for backend development, including the implementation of the map coloring puzzle solver algorithm.
3. SQLite/PostgreSQL: SQLite for development due to its simplicity; PostgreSQL is recommended for production on Heroku for robust data storage needs.
4. Flask-Login/Flask-JWT-Extended: For managing user sessions and authentication, with JWT for secure API communication.
5. Flask-SQLAlchemy: As the ORM tool for database operations, simplifying interactions between Python code and the database.
6. Heroku: For deploying the full-stack application, supporting both the Flask backend and React frontend.
7. Git: For version control, facilitating code management and Heroku deployments through Git integration.