Project Plan: Ghaziabad NH9 Cattle Problem - Prompt Engineering Game

Dev Kansara - Ahmedabad University
BTech Computer Science and Engineering (2027)

June 10, 2025

1. Introduction and Project Goal

This document outlines the development plan for an interactive web-based game designed to enhance students' prompt engineering skills. The game simulates the **"Ghaziabad NH9 Cattle Problem"** where students formulate AI prompts. A professor evaluates these prompts and moves the cows as per their effectiveness in solving the problem. The project will utilize **HTML**, **CSS**, and **JavaScript**.

2. Core Features

2.1. Student Interface

- Dedicated page for each student/team to enter their prompt.
- No visibility of other teams' prompts.

2.2. Professor Interface

- Centralized dashboard to view all submitted prompts from different teams.
- Ability to select a team's prompt for evaluation.
- Controls to move cow objects on the simulation page:
  - **"Good Prompt"** action: Moves selected cow(s) towards grass/off-road.
  - **"Bad Prompt"** action: Moves selected cow(s) towards the road.

2.3. Simulation Scene (Professor's View)

- Static HTML/CSS representation of NH9 and connecting highways in Ghaziabad.
- Multiple cow objects placed on the roads and adjacent green areas.
- Cow objects movable via JavaScript based on professor's input.

2.4. Team Management

- Ability to configure 2 to 6 teams (initial manual HTML setup, potential for dynamic creation).

3. Technical Breakdown and Implementation Plan

3.1. Frontend (HTML, CSS, JavaScript)

3.1.1. HTML Structure.

- `index.html`: Landing page for view selection.
- `student.html`: Form for prompt input, submit button, team identification.
- `professor.html`: Displays submitted prompts, prompt selection mechanism, evaluation buttons.
- `simulation.html`: Game canvas for cows (unique IDs), static scene elements (roads, green areas).

3.1.2. CSS Styling (`style.css`).

- Layout: Flexbox/Grid for responsive student/professor pages.
- Simulation Scene: Positioning of roads, grass, background elements; absolute positioning for movable cow images.
- Basic styling for interactive elements and thematic representation.

3.1.3. JavaScript Functionality.

- `script.js` (Student Side): Handles prompt submission, stores data in **localStorage** (e.g., `teamX_prompt`). Requires professor and student pages to be on the same origin.
- `professor_controls.js`: Loads prompts from `localStorage`, dynamically displays them. Event listeners for evaluation buttons. Communicates with `simulation.html`'s JavaScript (via `<iframe>`'s `contentWindow`) to trigger cow movement.
- `simulation_logic.js` (within `simulation.html`): Manages cow positions, implements `moveCowToGrass(cowId)` and `moveCowToRoad(cowId)` functions. Sets initial cow placements.

4. Development Stages

1. **Phase 1 (Days 1-2)**: Basic HTML & CSS Structure (`student.html`, `professor.html`, `simulation.html`, initial scene design).
2. **Phase 2 (Days 2-3)**: Core JavaScript - Cow Movement (`simulation_logic.js` functions).
3. **Phase 3 (Days 3-4)**: Prompt Submission & Display (`script.js` for `localStorage`, `professor_controls.js` for display).
4. **Phase 4 (Days 4-5)**: Professor Control & Simulation Integration (button event listeners, `<iframe>` communication).

5. **Phase 5 (Days 5-6)**: Refinement & Testing (add more cows, aesthetics, testing, instructions).

Optional Deliverables

- `index.html`, `student.html`, `professor.html`, `simulation.html`
- `style.css`, `script.js`, `professor_controls.js`, `simulation_logic.js`
- `Project README.md` with setup instructions