Trello Board Name: Sanskrit Gameshow (“Shabda Samvad”) MVP Development

Epic: Build Core Functionality for a Remote Sanskrit "Family Feud" Style Gameshow MVP

Description: This epic encompasses the development of the foundational components required to conduct a basic, single-round Sanskrit language gameshow with remote participants, focusing on survey data collection, a simplified game interface, and basic integration.

Features (Trello Cards within the Epic List):

List: MVP 0: Infrastructure Setup (Weeks 1-2)

- **Card Title:** Select Development Tools
  - **Description:** Research and choose the primary development tools (frameworks, libraries) for front-end and back-end development.
  - **Labels:** Technical, Planning
  - **Assignees:** [Assign Interns (groups of 5)] - *Scaled Assignment*
  - **Due Date:** [End of Week 1]
  - **Checklist:**
    - [ ] Evaluate front-end frameworks (React, Vue.js, Svelte).
    - [ ] Evaluate back-end frameworks (Node.js/Express, Python/Flask/Django).
    - [ ] Choose a database solution (SQLite, Google Sheets API, PostgreSQL).
    - [ ] Decide on a real-time communication library (Socket.IO).
    - [ ] Document the rationale for each tool selection.
- **Card Title:** Set Up Code Repository
  - **Description:** Create a repository (likely Git-based) for version control and collaboration.
  - **Labels:** Technical, DevOps
  - **Assignees:** [Assign Interns (groups of 5)] - *Scaled Assignment*
  - **Due Date:** [End of Week 1]
  - **Checklist:**
    - [ ] Choose a Git hosting service (GitHub, GitLab, Bitbucket).
    - [ ] Create a new repository for the project.
    - [ ] Set up initial project structure.
    - [ ] Configure basic repository settings (e.g., branch protection).
    - [ ] Create a README.md file.
- **Card Title:** Define Development Environment Setup
  - **Description**: Create a guide for interns to set up their local development environment.

    - **Labels**: Technical, DevOps
    - **Assignees**: [Assign Interns (groups of 5)] - *Scaled Assignment*
    - **Due Date**: [End of Week 2]
    - **Checklist:**
        - [ ] Document required software (Node.js, Python, etc.).
        - [ ] Provide instructions for installing dependencies.
        - [ ] Outline the process for cloning the repository.
        - [ ] Explain how to run the application locally.
        - [ ] Troubleshooting common setup issues.

List: Deliverable 1: Interactive Sanskrit Survey & Data Capture Module (Weeks 2-4)

- **Card Title:** Design Survey Creation Interface
    - **Description:** Develop a user-friendly web form for administrators to input Sanskrit survey questions. Include fields for the question text.
    - **Labels:** Front-End
    - **Assignees:** [Assign Interns (2 groups of 5)] - *Scaled Assignment*
    - **Due Date:** [End of Week 2]
    - **Checklist:**
        - [ ] Define input fields for questions.
        - [ ] Implement basic UI layout for question creation.
        - [ ] Ensure questions can be saved.
- **Card Title:** Implement Survey Submission Interface
    - **Description:** Create a web page where participants can view a survey question and input their answers.
    - **Labels:** Front-End
    - **Assignees:** [Assign Interns (2 groups of 5)] - *Scaled Assignment*
    - **Due Date:** [End of Week 3]
    - **Checklist:**
        - [ ] Design input field for participant answers.
        - [ ] Implement submission mechanism.
        - [ ] Basic UI for displaying the question.
- **Card Title:** Develop Back-End for Data Capture & Storage
    - **Description:** Set up the back-end logic to receive survey submissions and store the data (e.g., in SQLite or update Google Sheets).
    - **Labels:** Back-End, Database
    - **Assignees:** [Assign Interns (2 groups of 5)] - *Scaled Assignment*
    - **Due Date:** [End of Week 3]
    - **Checklist:**

        - [ ] Choose data storage method (SQLite/Google Sheets).
        - [ ] Implement API endpoint (if using a back-end framework) or Google Sheets API integration.
        - [ ] Ensure data is stored correctly linked to the question.
- **Card Title:** Implement Basic Answer Ranking Logic
    - **Description:** Write the code to retrieve the submitted answers for a given question and calculate the frequency of each unique answer to determine the initial ranking.
    - **Labels:** Back-End, Data Processing
    - **Assignees:** [Assign Interns (2 groups of 5)] - *Scaled Assignment*
    - **Due Date:** [End of Week 4]
    - **Checklist:**
        - [ ] Retrieve all answers for a specific question.
        - [ ] Count occurrences of each unique answer.
        - [ ] Sort answers by frequency (descending).
        - [ ] Display the ranked answers (basic output for now).
- **Card Title: Integration Testing - Deliverable 1**
    - **Description:** Create and execute integration tests for the survey and data capture module.
    - **Labels:** Testing, Integration
    - **Assignees:** [Assign Interns (1 group of 5)]
    - **Due Date:** [End of Week 4]
    - **Checklist:**
        - [ ] Write tests for survey creation.
        - [ ] Write tests for survey submission.
        - [ ] Write tests for data capture and storage.
        - [ ] Write tests for answer ranking logic.

List: Deliverable 2: Basic Remote Gameshow Interface Prototype (Weeks 5-8)

- **Card Title:** Design Host Interface (Single Round)
    - **Description:** Create a web interface for the host to display the current question and manually reveal answers and manage scores.
    - **Labels:** Front-End, UI/UX
    - **Assignees:** [Assign Interns (2 groups of 5)] - *Scaled Assignment*
    - **Due Date:** [End of Week 6]
    - **Checklist:**
        - [ ] Display the current question.
        - [ ] UI elements to "reveal" ranked answers (initially manual).
        - [ ] Input fields or controls to manually update team scores.

- **Card Title:** Design Participant Interface (Single Question)
  - **Description:** Develop a web interface for participants to view the current question and a simple mechanism to "buzz in."
  - **Labels:** Front-End, UI/UX
  - **Assignees:** [Assign Interns (2 groups of 5)] - *Scaled Assignment*
  - **Due Date:** [End of Week 7]
  - **Checklist:**
    - [ ] Display the current question.
    - [ ] Implement a "Buzz In" button or indicator.
    - [ ] Provide visual feedback when buzzed in (for the participant).
- **Card Title:** Implement Basic Real-Time "Buzz In" Signaling
  - **Description:** Use WebSockets (Socket.IO) to enable participants to signal the host when they want to answer.
  - **Labels:** Real-time, Back-End
  - **Assignees:** [Assign Interns (2 groups of 5)] - *Scaled Assignment*
  - **Due Date:** [End of Week 8]
  - **Checklist:**
    - [ ] Set up basic WebSocket connection.
    - [ ] Implement logic for a participant's "buzz" to be sent to the host.
    - [ ] Display on the host interface which participant buzzed in (basic indicator).
- **Card Title: Integration Testing - Deliverable 2**
  - **Description**: Create and execute integration tests for the gameshow interface.
  - **Labels**: Testing, Integration
  - **Assignees**: [Assign Interns (1 group of 5)]
  - **Due Date**: [End of Week 8]
  - **Checklist**:
    - [ ] Write tests for host interface.
    - [ ] Write tests for participant interface.
    - [ ] Write tests for buzz-in signaling.

List: Deliverable 3: Integration & Pilot Episode Simulation Tool (Weeks 9-12)

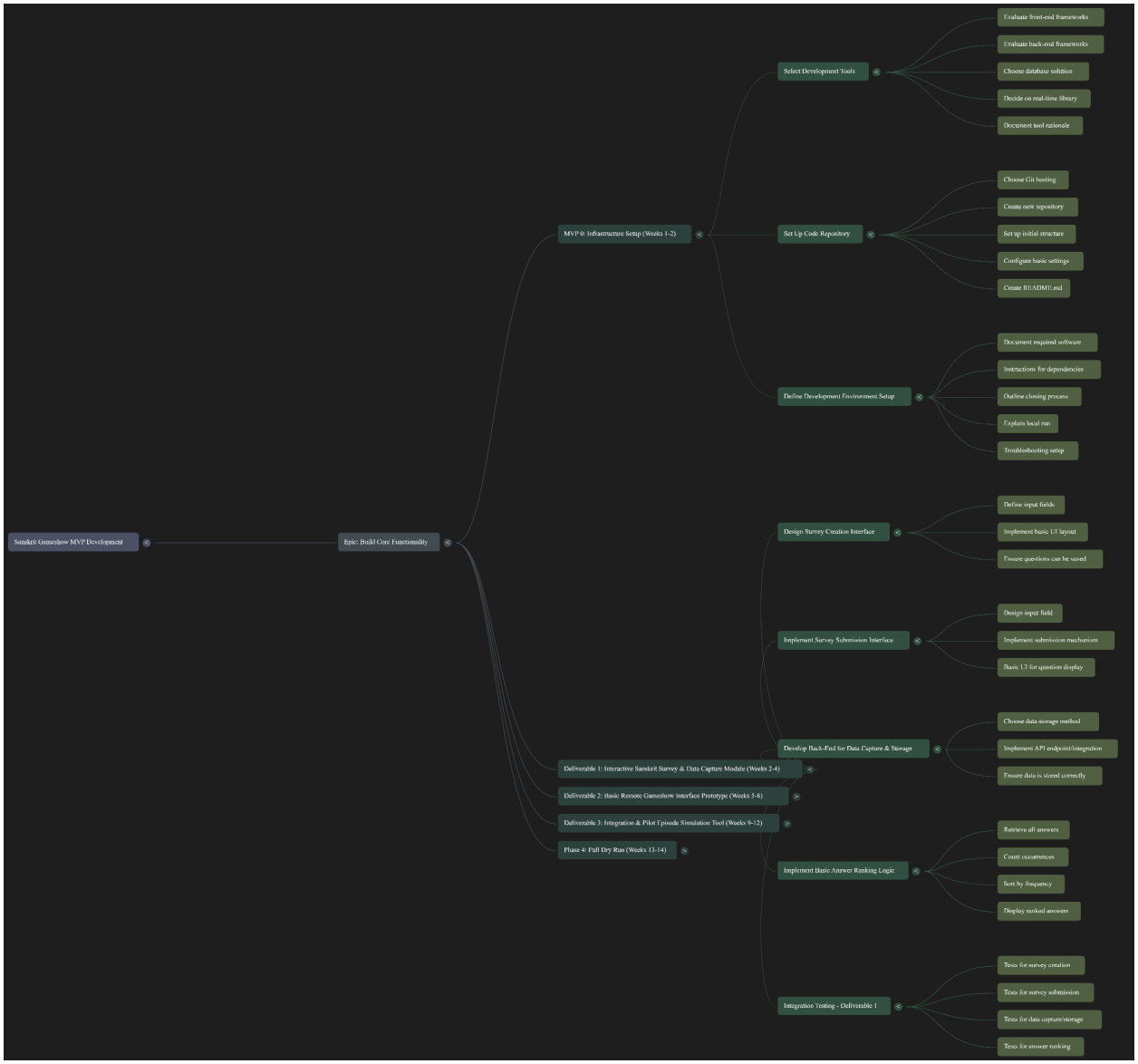
- **Card Title:** Integrate Survey Data with Host Interface
  - **Description:** Connect the host interface to the data generated in Deliverable 1, allowing the host to select a question and have the ranked answers (initially the top few) be available to reveal.
  - **Labels:** Integration, Back-End, Front-End
  - **Assignees:** [Assign Interns (2 groups of 5)] - *Scaled Assignment*

    - **Due Date:** [End of Week 10]
    - **Checklist:**
        - [ ] Implement API calls (if applicable) or data retrieval from Google Sheets.
        - [ ] Allow the host to select a question from the database.
        - [ ] Display the top ranked answers on the host interface (ready to be revealed).
- **Card Title:** Implement Basic Game Logic (Single Round)
    - **Description:** Add the core game logic for a single round, including handling correct/incorrect answers, updating scores, and revealing answers based on participant guesses.
    - **Labels:** Logic, Back-End, Front-End
    - **Assignees:** [Assign Interns (2 groups of 5)] - *Scaled Assignment*
    - **Due Date:** [End of Week 11]
    - **Checklist:**
        - [ ] Logic for the host to mark answers as correct or incorrect.
        - [ ] Automatic score updates based on correct answers.
        - [ ] Mechanism for revealing correct answers after a team's turn.
- **Card Title:** Create Pilot Episode Simulation Tool
    - **Description:** Refine the host interface to act as a basic simulation tool where the volunteer team can run through a single round of the game using the integrated data and logic.
    - **Labels:** Testing, Tooling, UI/UX
    - **Assignees:** [Assign Interns (2 groups of 5)] - *Scaled Assignment*
    - **Due Date:** [End of Week 12]
    - [ ] Streamline the host controls for running a simulation.
    - [ ] Ensure the basic game flow (question, buzz, answer, reveal, score) is functional.
    - [ ] Add basic instructions for using the simulation tool.
- **Card Title: Integration Testing - Deliverable 3**
    - **Description**: Create and execute integration tests for the pilot episode simulation.
    - **Labels**: Testing, Integration
    - **Assignees**: [Assign Interns (1 group of 5)]
    - **Due Date**: [End of Week 12]
    - **Checklist**:
        - [ ] Write tests for data integration.
        - [ ] Write tests for game logic.
        - [ ] Write tests for simulation tool.

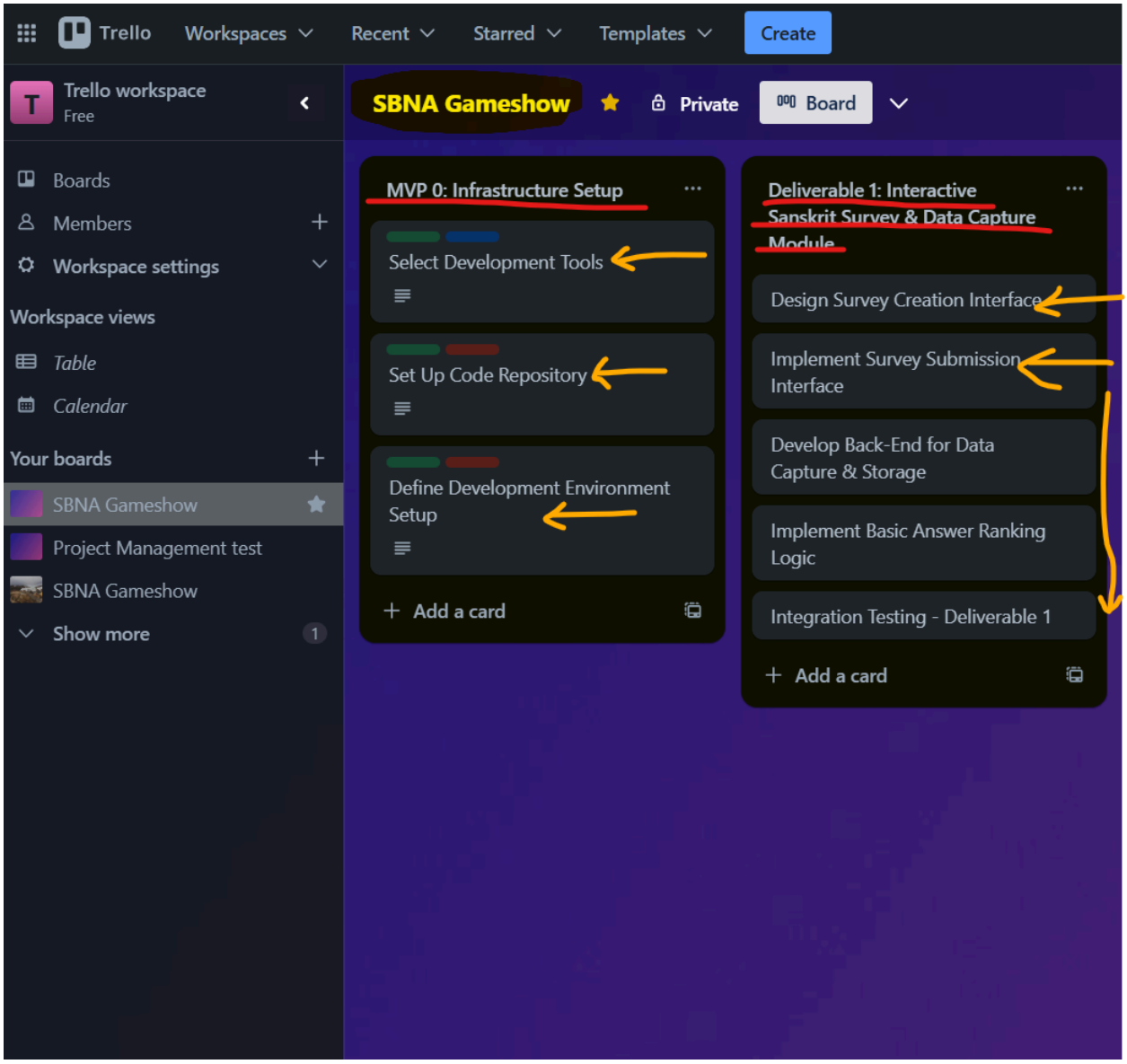
List: Phase 4: Full Dry Run (Weeks 13-14)

- **Card Title:** Conduct Full System Test
  - **Description:** Conduct a full test of the entire system with volunteer participants, simulating a complete game show episode.
  - **Labels:** Testing, Dry Run
  - **Assignees:** [All Interns]
  - **Due Date:** [End of Week 14]
  - **Checklist:**
    - [ ] Recruit volunteer participants.
    - [ ] Set up the remote game environment.
    - [ ] Run through a complete game show episode.
    - [ ] Document any issues or bugs.
    - [ ] Gather feedback from participants.
- **Card Title:** Finalize Documentation
  - **Description:** Complete all project documentation, including user manuals, technical documentation, and any lessons learned.
  - **Labels:** Documentation
  - **Assignees:** [Assign Interns (2 groups of 5)]
  - **Due Date:** [End of Week 14]
  - **Checklist:**
    - [ ] Complete user manual for the host.
    - [ ] Complete user manual for the participants.
    - [ ] Finalize technical documentation.
    - [ ] Create a lessons learned document.

MindMap Project Plan : NotebookLM Mind Map (4).png



Trello Board : https://trello.com/b/jytClZMo/sbna-gameshow


Trello
Workspaces
Recent
Starred
Templates
Create
Trello workspace
Free
Boards
Members
Workspace settings
Workspace views
Table
Calendar
Your boards
SBNA Gameshow
Project Management test
SBNA Gameshow
Show more
1
SBNA Gameshow
Private
Board
MVP 0: Infrastructure Setup
Select Development Tools
Set Up Code Repository
Define Development Environment Setup
Add a card
Deliverable 1: Interactive Sanskrit Survey & Data Capture Module
Design Survey Creation Interface
Implement Survey Submission Interface
Develop Back-End for Data Capture & Storage
Implement Basic Answer Ranking Logic
Integration Testing - Deliverable 1
Add a card