Project Name: Sanskrit Pictionary

Start Date: 2025-06-12

Estimated End Date:

End Date: 2025-10-10

Project Manager: Dr. Pradeep Dheerendra

Project Co-ordinator: Usman Khan

Product Owner: SamskritaBharati

🛠️ Development Team Structure

Initial Setup

- **The game was originally developed by 3 separate teams:**

    - **Backend**

    - **Frontend**

    - **UI/UX**

Going Forward

- **All team members will:**

    - **Gain understanding of both frontend & backend.**

    - **Be assigned tasks based on remaining work, not restricted by original team roles.**

✨ Project Overview

Project Name: Sanskrit Pictionary — "Sketch, Guess, Learn!".

Project Title: Online Single/Multiplayer Guessing game (inspired by Pictionary)

Project Category : TA/ET

Project Description: A multiplayer drawing and guessing game designed to help players learn Sanskrit vocabulary in a fun, interactive way. Players take turns drawing words while others guess in real time. After each round, flashcards are shown with Devanagari script, Latin transliteration, English translation, and audio — providing reinforcement without compromising gameplay fun.

While traditional flashcard apps focus purely on memorization, this game will *blend retention with engagement* by layering:

- **Flashcard reveals** (after and optionally before each round) including:
  - Devanagari script
  - Latin transliteration
  - Audio pronunciation
  - Image
  - English meaning
- **Optional Enhancements**
  - AI-generated clues (using OpenAI/Cohere APIs)
  - Like/dislike ratings on drawings
  - Word history with replayable audio/visuals

This keeps the game fun and active while sneaking in *serious language learning* in the background.

🔍 Target Users

- Beginners learning Sanskrit

- Educators incorporating edtech

- Culturally curious casual gamers

🚀 Tech Stack

- **Frontend**: React.js + CSS
- **Canvas Drawing**: React Sketch Canvas
- **Real-time Communication**: Node.js + Express + Socket.IO
- **Backend**: MongoDB (MERN stack)
- **Audio**: HTML5 `<audio>` or preloaded files
- **GitHub**: For Version Control and storage.
- **AI (Optional)**: OpenAI/Cohere APIs for hints or translations

---

🎮 Core Game Flow (MVP)

1. Welcome & Entry (Sprint 1)

- Players land on **Welcome Page**.

- Options: **Guest Play** or **Sign Up / Log In**.

- Access **Rules** & **Tutorial**.

- Click **Play Now** → generates **lobby link**.

---

2. Game Setup (Lobby Page) (Sprint 2–3)

- Players join via **shared link**.

- ✅ Game requires **minimum 4 players** to begin.

- Display:

    - List of joined players (avatars + scores).

- Optional (future):

    - Customize rounds, timer, difficulty.

    - Solo mode.

    - Play against system.

---

3. Game Rounds

Drawer Selection & Flashcard Reveal

- System randomly picks 1 drawer.

- Drawer sees flashcard in **stages**:

    1. Devanagari script + image

    2. Latin transliteration (delayed or on click)

    3. Audio pronunciation

    4. (Optional) English translation (drawer loses half points if revealed)

Drawing Phase

- Drawer sketches the word.

- No writing or direct hints allowed.

Guessing Phase

- Guessers type answers in **Harvard-Kyoto transliteration**.

- Real-time drawing updates.

- If time runs out or guessers forfeit → system reveals **English meaning** + audio.

Scoring & Recap

- Points: based on **speed & accuracy**.

- Recap:

    ○ Flashcard shown to all.

    ○ Correct guessers listed in order.

    ○ Players give **like/dislike feedback**.

    ○ Word saved in **Word History panel**.

---

4. End Game

- Final **leaderboard** displayed.

- Highlight **most-liked drawing**.

- Options:

  - Replay with same players.

  - Start new game (new lobby).

  - Quit (player leaves, rest can continue).

---

5. Rules Summary

- ❌ No writing the word directly.

- ❌ No revealing letters explicitly.

- ✅ Only drawing, miming, or visual clues allowed.

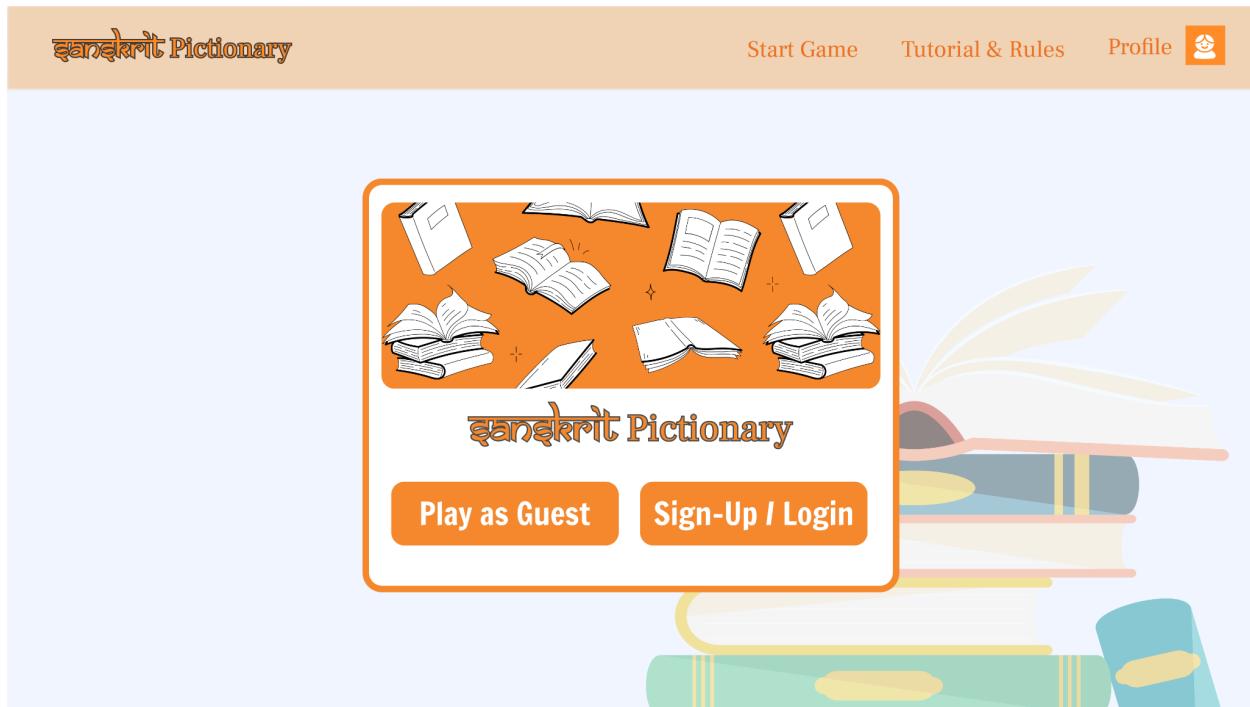- ✅ Scoring based on guessing speed & accuracy.

📌 Optional / Future Features

- ✅ Solo mode (player vs random users).

- ✅ PDF export of word history (guest) / permanent history (registered user).

- ✅ Difficulty levels.
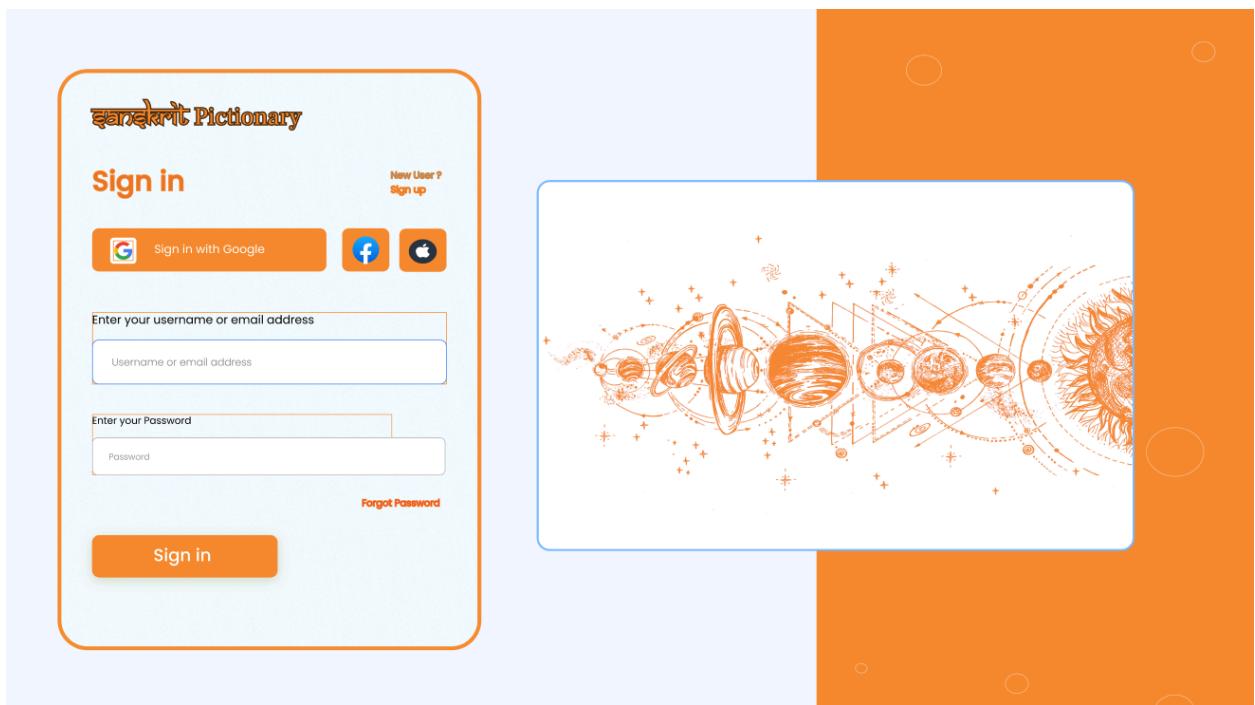
- ✅ Timer customization.
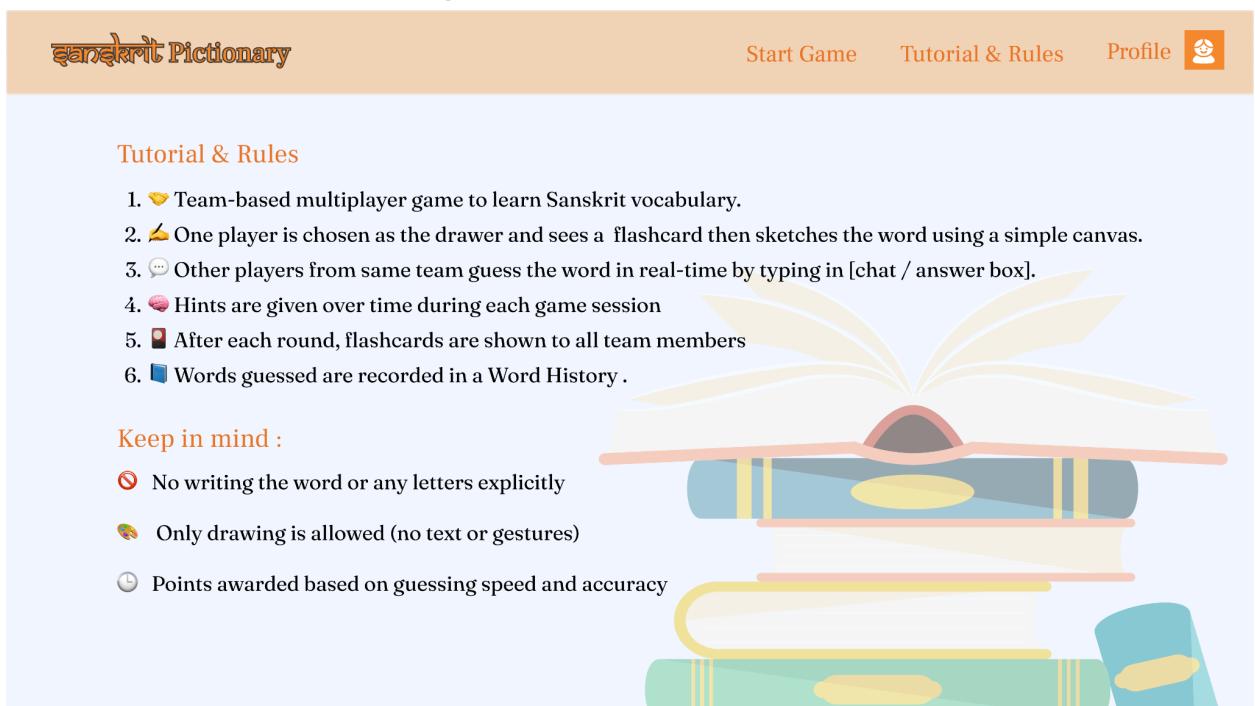
WireFrames

1. Welcome/Landing Page

2. Sign Up Page

3. Login Page

संस्कृत Pictionary

Sign in

New User ?
Sign up

[G] Sign in with Google [f] [🍎]

Enter your username or email address

Username or email address

Enter your Password

Password

Forgot Password

Sign in

4. Tutorial & Rules Page

संस्कृत Pictionary   Start Game   Tutorial & Rules   Profile [👤]

Tutorial & Rules

1. 🤝 Team-based multiplayer game to learn Sanskrit vocabulary.
2. ✍️ One player is chosen as the drawer and sees a flashcard then sketches the word using a simple canvas.
3. 💬 Other players from same team guess the word in real-time by typing in [chat / answer box].
4. 🧠 Hints are given over time during each game session
5. 🎴 After each round, flashcards are shown to all team members
6. 📘 Words guessed are recorded in a Word History .

Keep in mind :

🚫 No writing the word or any letters explicitly

🎨   Only drawing is allowed (no text or gestures)

🕐 Points awarded based on guessing speed and accuracy

5. Lobby

Sanskrit Pictionary

Start Game   Tutorial & Rules   Profile

Game Lobby URL: https://www.shareableULR.com   🔗 Copy Link

User List

😀	Alice	Score: 0
🙂	Bob	Score: 0
😎	Charlie	Score: 0

Game Settings

Select Rounds
Choose number of rounds
1 2 3 4 5 6

Select Timer
Duration of each round
30 45 60 75 90

Select Difficulty
Adjust the challenge level
Easy Medium Hard

Start Game

6. Game Panel

Sanskrit Pictionary

Start Game   Tutorial & Rules   Profile

Time Left : 50 sec   **Word Hint :**   P _ S _ _ K   **Your score : 20**

Players

- Sera
- Julia
- Hannah
- Samantha
- Joy

Drawing Board

↻ Restart

पुस्तक | Pustak
(HINT)

Start Drawing

Chat Box

Say Something Nice ➤

Word History

Sanskrit	Sanskrit	Sanskrit	Sanskrit	Sanskrit
English	English	English	English	English

7. Scores Display

Start Game   Tutorial & Rules   Profile

Full Leanderboard

1st

Sera

Score : 70

2nd

Julia

Score : 50

3rd

Joy

Score : 40

Leaderboard

Best Drawing

New Game   Exit to Home

📊 Database Models - refer to code

Name	Gmail Address(for Agile setup)	School-Email	GitHub	College	Branch Name	Game is running locally
Usman Khan	u.khan980212@gmail.com	khan356@sheridancollege.ca	https://github.com/UsmanKhanDeveloper	Sheridan - Trafalgar Campus	Npm-install -despair	yes
					fall-term-refactor	

📅 **Internship Action Plan – Game Project (Now → Mid December)**

🎯 **Overall Goal**
Stabilize, polish, and deploy the game during the internship period. Focus areas:

- Team onboarding & shared understanding of the codebase.

- Enhancing the gameplay session experience.

- Fixing critical bugs and improving stability.

- Preparing the codebase for long-term maintainability (SRP refactor).

- Setting up AWS-based CI/CD pipelines and testing deployments beyond localhost.

- (Optional) OpenAI API integration for gameplay enhancement.

Action Plan

1. Team Onboarding & Documentation

- Ensure all teammates can set up and run the game locally.

- Maintain shared Google Doc/Notion with setup notes, architecture guides, and refactor logs.

- Create and update the "Codebase Trace & Refactor Guide" for clarity.

2. Refactor & Codebase Improvements

- Apply SRP refactor: break large files into smaller, modular components.

- Document each module's responsibilities and ownership.

- Continuously refactor as new features and fixes are introduced.

3. Gameplay Enhancements & Stability

- Refactor and stabilize core flows (Welcome → Lobby → Gameplay → End Game).

- Fix critical issues such as:

  - Solo mode matchmaking (random online users).

  - Clearing drawing data after each turn switch.

  - Extending team-switch popup display for smoother gameplay transitions.

- Add UX improvements (flashcard reveal, scoring recap, smoother session flow).

4. AWS CI/CD & Deployment

- Set up AWS pipelines for continuous integration and deployment.

- Transition deployment from localhost → staging → production.

- Test multiplayer sessions, lobby formation, and game stability under real conditions.

5. Finalization & Optional Enhancements

- Wrap up with clear documentation of fixes, refactor learnings, and CI/CD setup.

- Optional: explore OpenAI API for gameplay improvements (if timeline permits).

📄 Sanskrit Pictionary – Codebase Trace & Refactor Guide