**🐾 Virtual Pet Simulator – Project Report**

📌 **Project Title:**

**Virtual Pet Simulator using HTML, CSS, and JavaScript**

👨‍💻 **Developed By:**

Rohit Reddy Mandadi

(Unified Mentor - Frontend Developer Intern)

📝 **Abstract:**

The Virtual Pet Simulator is a fun and interactive web application that simulates the experience of caring for a digital pet. Built using HTML, CSS, and JavaScript, the simulator allows users to select a pet (cat, dog, or bunny) and interact with it by feeding, playing, and resting it. The pet reacts with changing moods based on its stats: hunger, happiness, and energy.

🎯 **Objectives:**

* To build an engaging front-end web project.
* To understand state management using JavaScript.
* To apply DOM manipulation for interactivity.
* To integrate sound effects for user experience.
* To allow avatar-based customization (cat, dog, bunny).

🛠️ **Technologies Used:**

* HTML5 – Structure of the application.
* CSS3 – Styling, layout, and visual feedback.
* JavaScript (Vanilla) – Logic, state management, interactivity.
* Sound Effects (MP3) – For feedback on user actions.

**🧩 Features:**

**Feature Description**

🐾 Pet Avatar Selection Choose between cat, dog, or bunny.

📊 Status Bars Live stats for Hunger, Happiness, and Energy.

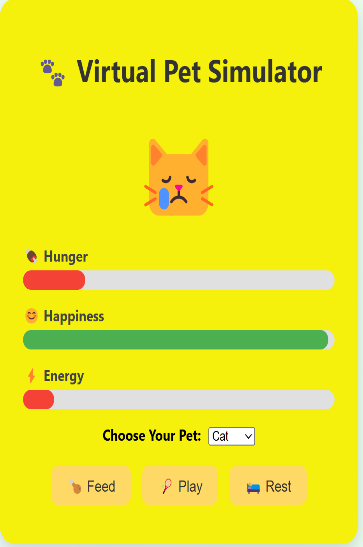
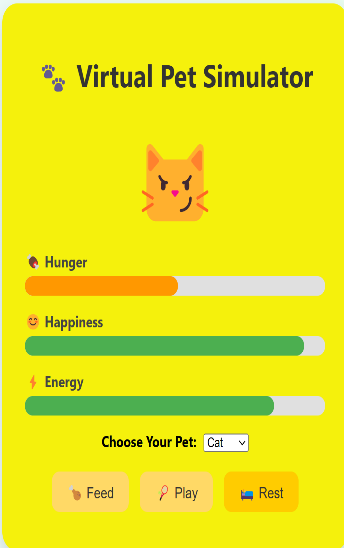
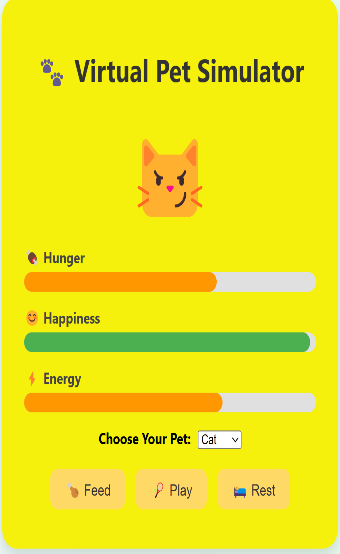
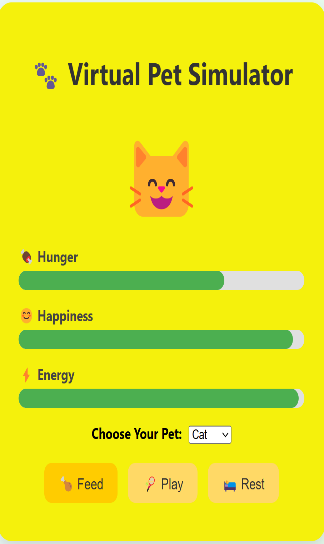
🎮 Action Buttons Feed, Play, and Rest – each affects pet stats.

😸 Mood Reactions Pet emoji changes based on mood (happy, sad, neutral).

🔊 Sound Effects Plays unique sounds on actions.

🎨 Responsive Design Works well on desktop and mobile screens.

**🎨 UI Snapshot**



**🔁 Working Logic:**

* Feed increases hunger and energy.
* Play increases happiness but decreases energy and hunger.
* Rest increases energy but decreases hunger.
* The pet’s emoji changes based on its mood calculated from all three stats.
* Stats are visualized using progress bars with dynamic color changes.

**🌟 Future Enhancements:**

* Add pet naming feature.
* Store progress in localStorage.
* Add custom pet avatars (images or animated SVGs).
* Introduce a timer-based stat decay for real-time challenge.
* Enable mobile app version using PWA (Progressive Web App).

**📁 File Structure:**

cpp

Copy

Edit

virtual-pet-simulator/

│

├── **index.html**

├── **style.css**

├── **script.js**

└── **sounds/**

├── **feed.mp3**

├── **play.mp3**

└── **rest.mp3**

**📌 Conclusion:**

This project showcases the power of front-end development in creating interactive and user-centric applications. It combines creativity with logic and demonstrates an understanding of DOM manipulation, state-driven UI, and responsive design — making it an ideal project for a portfolio or as a fun learning experience.