Here’s a breakdown of **2-week Agile sprints** from January 1 to April 4 for your SPA project. Each sprint has specific goals and deliverables, following the Agile methodology. The focus is on building a Flash Card Creator using **HTML**, **CSS**, and **JavaScript**, with JSON for storing and pulling data.

**Sprint 1: Jan 1 – Jan 14**

**Goal**: Project Initialization and Basic Setup

1. **Tasks**:
   * Create a GitHub repository and initialize the project.
   * Set up the basic folder structure (HTML, CSS, JS, assets, etc.).
   * Design a simple homepage wireframe (pen-and-paper or tool like Figma).
   * Build a static homepage using HTML and CSS.
   * Write a project README file with goals and tech stack.
2. **Deliverables**:
   * Static homepage with a simple layout.
   * Basic GitHub repository with an initial commit.

**Sprint 2: Jan 15 – Jan 28**

**Goal**: User Interface for Flash Card Creation

1. **Tasks**:
   * Create a form for users to input flashcard details (question, answer, and optional category).
   * Style the form using CSS for a clean, responsive design.
   * Add JavaScript to capture user input and store it temporarily in a JSON-like object.
2. **Deliverables**:
   * A functional form that collects flashcard data and stores it in memory (JavaScript object).
   * Basic validation for required fields (e.g., no empty questions/answers).

**Sprint 3: Jan 29 – Feb 11**

**Goal**: JSON File Integration for Data Storage

1. **Tasks**:
   * Implement functionality to save user-created flashcards into a JSON file (use Blob for downloads).
   * Allow users to upload a JSON file to load saved flashcards.
   * Create a simple error-handling mechanism for incorrect JSON formats.
2. **Deliverables**:
   * Users can save flashcards as a downloadable JSON file.
   * Users can upload a JSON file to populate flashcards in the app.

**Sprint 4: Feb 12 – Feb 25**

**Goal**: Display Flash Cards and Basic Interactivity

1. **Tasks**:
   * Create a section to display flashcards dynamically from the loaded JSON data.
   * Add a “flip” animation to view the answer when clicking on a flashcard.
   * Style the flashcards for consistency and readability.
2. **Deliverables**:
   * Flashcards displayed dynamically on the page.
   * Interactive flip animation for flashcards.

**Sprint 5: Feb 26 – Mar 10**

**Goal**: Flash Card Navigation and Filters

1. **Tasks**:
   * Add navigation controls to move between flashcards (e.g., next/previous buttons).
   * Implement category-based filtering of flashcards.
   * Style navigation controls and filtering options.
2. **Deliverables**:
   * Users can navigate between flashcards.
   * Users can filter flashcards by category.

**Sprint 6: Mar 11 – Mar 24**

**Goal**: Advanced Features and LocalStorage

1. **Tasks**:
   * Implement LocalStorage to persist user data between sessions (e.g., save flashcards in the browser).
   * Add a “study mode” with shuffled flashcards for random practice.
   * Create a button to reset all data (clear flashcards and LocalStorage).
2. **Deliverables**:
   * Flashcards persist between sessions using LocalStorage.
   * Study mode with shuffled flashcards.

**Sprint 7: Mar 25 – Apr 4**

**Goal**: Testing, Polishing, and Final Delivery

1. **Tasks**:
   * Perform user testing to identify and fix bugs.
   * Optimize performance and clean up unused code.
   * Polish the UI/UX (responsive design, animations, accessibility).
   * Update the README file with setup instructions and screenshots.
2. **Deliverables**:
   * A polished and fully functional Flash Card Creator SPA.
   * Finalized GitHub repository with detailed documentation.

**Key Considerations:**

* **Weekly Standups**: Check progress every week and adjust tasks as needed.
* **Retrospectives**: At the end of each sprint, review what worked well and what could improve.
* **Future Extensions** (optional for post-April):
  + Add user authentication.
  + Sync data to a backend (e.g., Node.js, Firebase).
  + Add themes (light/dark mode).