

# TamilTheni Product Requirements Document

**Version:** 1.0

**Last Updated:** January 2026

**Status:** Active Development

---

## Table of Contents

1. Product Overview
  2. Business Requirements
  3. Functional Requirements
  4. Technical Requirements
  5. Non-Functional Requirements
  6. Module-Specific Requirements
  7. Data Requirements
  8. Workflow & Automation Requirements
  9. Quality Assurance Requirements
  10. Deployment Requirements
- 

## Product Overview

### Purpose

TamilTheni is a Tamil language learning web application designed to help students prepare for the **FETNA Tamil Theni Competition** (<https://tamiltheni.org/>). The application provides interactive study modules covering vocabulary, sentence construction, translation, and word discovery skills.

### Target Audience

- Tamil school students (primarily in the USA)
- Competition participants for FETNA Tamil Theni 2026
- Tamil language learners

### Stakeholders

- Peoria Tamil School (Development Team)
  - FETNA (Competition organizer)
  - Students and Parents
- 

## Business Requirements

### BR-001: Competition Alignment

The application must align with the official FETNA Tamil Theni competition format and word lists.

Attribute	Value
Priority	Critical
Status	Implemented

**BR-002: Free and Accessible**

The application must be freely accessible to all Tamil school students without requiring registration or payment.

Attribute	Value
Priority	Critical
Status	Implemented
Implementation	GitHub Pages hosting

**BR-003: Five Competition Modules**

The application must support all five categories of the Tamil Theni competition.

Module	Competition Format	Description
Theni 1	Picture → Tamil Word	Identify the Tamil word for a displayed picture
Theni 2	Picture Pair → Sentence	Form a Tamil sentence using two displayed pictures
Theni 3 & 4	English → Tamil Translation	Translate English sentences to Tamil
Theni 5	Clue Words → Target Word	Find the target word using clue words

Attribute	Value
Priority	Critical
Status	Implemented

**BR-004: Difficulty Levels**

The application must support two difficulty levels matching the competition structure.

Level	Description
D1	Easier words, suitable for younger students
D2	Advanced words, for older students

Attribute	Value
Priority	High
Status	Implemented

**BR-005: Category Filtering**

Users must be able to filter content by word categories (Body Parts, Food, Animals, etc.).

Attribute	Value
Priority	High

Attribute	Value
Status	Implemented

### BR-006: Progressive Web App (PWA)

The application should be installable on mobile devices and work offline.

Attribute	Value
Priority	Medium
Status	Implemented

## Functional Requirements

### FR-001: Navigation System

- ☐ First/Previous/Next/Last navigation buttons
- ☐ Keyboard navigation (Arrow keys, Home, End)
- ☐ Progress counter showing current position
- ☐ Progress bar visualization

### FR-002: Timer Functionality

- ☐ Configurable countdown timer per module
- ☐ Visual timer display (circular pie-chart style)
- ☐ Timer toggle checkbox in settings
- ☐ Auto-advance option when timer expires
- ☐ Alarm sound on timer completion

Module	Default Timer Duration
Theni 1	10 seconds
Theni 2	20 seconds
Theni 3 & 4	15 seconds
Theni 5	60 seconds

### FR-003: Audio/Text-to-Speech

- ☐ Text-to-Speech for English words (Theni 1)
- ☐ Audio toggle checkbox to enable/disable
- ☐ Consecutive audio playback on slide changes
- ☐ Browser autoplay policy compliance (require user gesture)

### FR-004: Shuffle and Reset

- ☐ Shuffle button to randomize slide order
- ☐ Reset button to restore original sequence

### FR-005: Control Panel

- ☐ Collapsible settings panel
- ☐ Category selection dropdown
- ☐ Difficulty filter buttons (All/D1/D2)
- ☐ Timer and audio toggles

### **FR-006: Reveal Mechanism (Theni 3 & 4)**

- ☐ Hidden answer that reveals on click/tap
  - ☐ Click-to-flip card interaction
- 

## **Technical Requirements**

### **TR-001: Build System**

- ☐ Vite as the build tool
- ☐ TypeScript for all application logic
- ☐ Hot Module Replacement (HMR) in development
- ☐ Minified production builds

### **TR-002: TypeScript Architecture**

- ☐ Strong typing for all data interfaces
- ☐ Shared utility modules (`utils.ts`, `timer.ts`, `audio_manager.ts`)
- ☐ Layout module for consistent UI injection (`layout.ts`)
- ☐ Centralized configuration (`config.ts`)

### **TR-003: CSS Architecture**

- ☐ Design tokens in `tokens.css`
- ☐ Shared styles in `style.css`
- ☐ Module-specific CSS files
- ☐ Responsive design for mobile/tablet

### **TR-004: Data Format**

- ☐ JSON data files in `src/data/`
- ☐ Single source of truth for word lists
- ☐ Merged data for Theni 1 & 2 (`theni_words.json`)
- ☐ Separate data for Theni 5 (`theni5_words.json`)

### **TR-005: Image Handling**

- ☐ Local images in `public/assets/images/theni12/`
- ☐ Filename convention: `{word_en}.jpg`
- ☐ Fallback to placeholder on image load failure
- ☐ Image caching to avoid re-fetching

### **TR-006: Browser Compatibility**

- ☐ Modern browsers (Chrome, Firefox, Safari, Edge)
- ☐ Mobile browsers (iOS Safari, Chrome Mobile)
- ☐ Web Speech API support for TTS

### **TR-007: Code Quality**

- ☐ ESLint for linting
- ☐ Prettier for formatting
- ☐ Type checking via `tsc`

## TR-008: Cache Management

- ☐ Vite handles cache invalidation via content hashing
  - ☐ Service worker for offline caching (PWA)
  - ☐ ~~Bump version script~~ (Deprecated after Vite migration)
- 

## Non-Functional Requirements

### NFR-001: Performance

Metric	Target
Initial page load	< 3 seconds
Slide navigation	< 100ms
Image load time	< 1 second per image
Build time	< 30 seconds

### NFR-002: Accessibility

- ☐ Keyboard navigation support
- ☐ Screen reader compatibility
- ☐ High contrast text for readability
- ☐ Touch-friendly button sizes

### NFR-003: Reliability

- ☐ Graceful degradation on network failures
- ☐ Error handling for missing images
- ☐ Console error logging for debugging

### NFR-004: Maintainability

- ☐ Modular code structure
- ☐ Separated concerns (HTML/CSS/TS/Data)
- ☐ Documented architecture (ARCHITECTURE.md)
- ☐ Clear file naming conventions

### NFR-005: Scalability

- ☐ Support for 800+ words
  - ☐ Ability to add new categories
  - ☐ Extensible module structure
- 

## Module-Specific Requirements

### Theni 1: Picture to Tamil Word

#### T1-001: Display Requirements

- ☐ Single word card per slide
- ☐ Image prominently displayed
- ☐ English word shown
- ☐ Tamil word (revealed on demand)
- ☐ Category and difficulty badges

### **T1-002: Audio Requirements**

- ☐ Auto-pronounce English word on slide change
- ☐ Audio toggle in settings
- ☐ Console audio playback on consecutive slides

### **T1-003: Image Requirements**

- ☐ High-quality, relevant images
  - ☐ Localized images where possible
  - ☐ Fallback placeholder for missing images
- 

## **Theni 2: Sentence Building with Pictures**

### **T2-001: Display Requirements**

- ☐ Dual-card layout (two words side-by-side)
- ☐ Random pairing of words
- ☐ Persistent partner pairing per session
- ☐ Both English and Tamil words

### **T2-002: AI Sentence Generation**

- ☐ Gemini AI integration for sentence generation
- ☐ API key input in settings panel
- ☐ Generated sentence caching
- ☐ Tamil sentence with English translation

### **T2-003: Virtual Slides**

- ☐ Virtual DOM elements for filtering (no physical DOM appending)
  - ☐ Proper HTML template formatting (querySelector compatibility)
- 

## **Theni 3 & 4: Translation Practice**

### **T34-001: Display Requirements**

- ☐ English sentence prominently displayed
- ☐ Tamil translation hidden initially
- ☐ Click-to-reveal interaction
- ☐ Target word highlighted in both languages

### **T34-002: Sentence Quality**

- ☐ Context-appropriate sentences
  - ☐ Grade-appropriate complexity
  - ☐ Accurate translations
- 

## **Theni 5: Word Discovery**

### **T5-001: Display Requirements**

- ☐ Clue words displayed prominently
- ☐ Answer hidden initially

- ☐ Click-to-reveal mechanism
- ☐ Word meaning shown after reveal

#### T5-002: Timer Requirements

- ☐ 60-second default timer
  - ☐ Circular timer visualization
  - ☐ Auto-start option when timer is enabled
- 

## Data Requirements

#### DR-001: Word Data Schema

```
{
  "id": "number (unique)",
  "category": "string (English)",
  "category_ta": "string (Tamil)",
  "difficulty": "D1 | D2",
  "word_en": "string",
  "word_ta": "string",
  "image_word": "string (filename base)",
  "sentence_en": "string (HTML with <b> tags)",
  "sentence_ta": "string (HTML with <b> tags)",
  "complexity": "number (1-3)"
}
```

#### DR-002: Data Volume

Dataset	Count
Theni 1 & 2 words	~800
Theni 5 clues	~50
Categories	20+
Images	~800

#### DR-003: Image Requirements

Attribute	Requirement
Format	JPEG
Resolution	300x180 minimum
Quality	Clear, relevant to word
Naming	Lowercase, underscore-separated

---

## Workflow & Automation Requirements

#### WA-001: Image Verification Pipeline

- ☐ Script to verify all words have corresponding images
- ☐ Automated fallback image generation
- ☐ Image quality validation

### **WA-002: Data Processing Pipeline**

- ☐ PDF to JSON conversion for FETNA word lists
- ☐ Data augmentation scripts (sentences, translations)
- ☐ Validation scripts for data integrity

### **WA-003: Development Workflow**

- ☐ `npm run dev` - Development server with HMR
  - ☐ `npm run build` - Production build
  - ☐ `npm run preview` - Preview production build
  - ☐ `npm run lint` - Code linting
  - ☐ `npm run format` - Code formatting
  - ☐ `npm test` - Run unit/integration tests
- 

## **Quality Assurance Requirements**

### **QA-001: Unit Testing**

- ☐ Vitest for unit tests
- ☐ Test coverage for utility functions
- ☐ Test coverage for timer module

### **QA-002: Build Acceptance Tests (BAT)**

- ☐ BAT tests for each module initialization
- ☐ Filter functionality tests
- ☐ Navigation tests
- ☐ Card reveal logic tests

### **QA-003: Manual Testing Checklist**

- ☐ Verify all modules load correctly
  - ☐ Test on multiple browsers
  - ☐ Test on mobile devices
  - ☐ Verify audio functionality
  - ☐ Check image loading
- 

## **Deployment Requirements**

### **DP-001: Hosting**

- ☐ GitHub Pages deployment
- ☐ Custom domain support (tamiltheni.org)
- ☐ HTTPS enabled

### **DP-002: Build Artifacts**

- ☐ Output to `docs/` folder
- ☐ Minified JavaScript bundles
- ☐ Optimized CSS
- ☐ Copied static assets



### DP-003: Branching Strategy

---

Branch	Purpose
publish	Production-ready code, deployed to GitHub Pages
main	Development branch

---

---

### Issue Tracking (Resolved)

Issue	Description	Status
Theni 2 Display	Words and images not showing	Fixed (malformed HTML template)
Consecutive Audio	Audio not playing on slide changes	Fixed
Browser Hung Processes	Test processes hanging	Fixed (process cleanup)
Vite Cache	Cache invalidation after migration	Resolved (Vite handles automatically)

---

---

### Change Log

---

Date	Version	Changes
Jan 2026	1.0	Initial requirements document created

---

---

*This document is maintained alongside the codebase and should be updated as requirements evolve.*