Product Requirements Document (PRD) for Make Story Time 2.0

Overview

Make Story Time 2.0 is a web application that allows users to create personalized stories. The application leverages modern AI technology to generate custom stories based on character information, converts text to audio, and provides a beautiful reading experience with downloadable formats. The site runs currently at makestorytime.com.

Core Features

1. Story Creation

- Users can initiate the creation of a new custom story
- Multiple story templates and genres available for selection
- Character-centered approach for personalization rather than directly collecting child information:
    - Character name
    - Character age
    - Character traits (curious, brave, shy, etc.)
    - Favorite things (activities, animals, etc.)
    - Special abilities or interests
- Target audience selection to adjust the tone and complexity:
    - Young readers (ages 3-8)
    - Middle grade readers (ages 9-12)
    - Teen readers
    - Adult readers
- Setting options (time and place)
- Theme, conflict, and moral selection
- Special requests section for additional customization
- AI-generated stories personalized to the specifications

2. Payment Processing

- $15 per story via Stripe Checkout
- Secure payment flow
- Order confirmation email

3. AI Integration

- OpenRouter API integration for story generation
- Text-to-speech API for audio narration
- Contextual prompting based on character information

4. Story Display & Downloads

- Immersive native book interface for reading the story with optimized typography
- Mobile-first design for all reading and listening experiences
- Integrated audio player for listening to narration while viewing the story
- Prominent download options for both PDF and MP3 formats
- Full-screen reading mode with page-turning animations
- Responsive text sizing for different devices

5. User Dashboard

- Library of all created stories
- Quick access to read or download each story
- Creation date and basic metrics

6. Future Expansion

- Print-on-demand integration (Printful)
- Ability to customize story aspects (theme, length, style)
- Gift options and sharing features
- Offline reading capabilities using service workers
- Subscription model for regular story creation
- Enhanced illustrations based on story content
- Multiple narration voice options
- Interactive elements within stories

Technology Stack

- Backend: Laravel 12
- Frontend: Blade templates with Tailwind CSS 4
- Interactivity: Livewire
- Database: SQLite (chosen for simplicity, easy deployment, and sufficient performance for expected load)
- Storage: Cloudflare R2
- Payment: Stripe
- APIs: OpenRouter (AI), Text-to-Speech service

User Interface

- Mobile-first design throughout the entire application
- Clean, intuitive step-by-step form approach similar to StoryAI
- Genre selection with visual icons
- Character creation form optimized for touch input
- Setting and theme inputs with visual cues
- Special requests section
- Progress indicators for form completion
- Immersive book reading experience with:
    - Page-turning animations
    - Background textures/themes
    - Text that adapts to screen size
    - Optional read-along highlighting
    - Synchronized audio narration
- Persistent audio player that allows users to continue listening while navigating

Entity Relationship Diagram (ERD)

USERS
- id (PK)
- name
- email
- password
- stripe_customer_id
- created_at
- updated_at

STORIES
- id (PK)
- user_id (FK -> USERS.id)
- story_template_id (FK -> STORY_TEMPLATES.id)
- title
- content
- audio_path
- target_audience (enum: young, middle_grade, teen, adult)
- status (enum: pending, processing, generated, completed, failed)
- metadata (JSON)
- created_at
- updated_at

CHARACTERS
- id (PK)
- story_id (FK -> STORIES.id)

- name
- age
- traits
- favorites
- special_abilities
- is_main_character (boolean)
- created_at
- updated_at

STORY_TEMPLATES
- id (PK)
- title
- description
- prompt_template
- genre
- metadata (JSON)
- created_at
- updated_at

STORY_SETTINGS
- id (PK)
- story_id (FK -> STORIES.id)
- time_period
- location
- theme
- conflict
- moral
- special_requests
- created_at
- updated_at

PAYMENTS
- id (PK)
- user_id (FK -> USERS.id)
- story_id (FK -> STORIES.id)
- amount
- stripe_payment_id
- status (enum: pending, completed, failed, refunded)
- created_at
- updated_at

Relationships

- A User can have many Stories
- A Story belongs to a User
- A Story can have many Characters
- A Story has one StorySettings
- A Story is based on a StoryTemplate
- A Payment belongs to a User and a Story

User Flow

1. User registers/logs in
2. User navigates to "Create Story" page
3. User selects genres and story template
4. User creates main character and adds details
5. User defines setting (time and place)
6. User adds theme, conflict, and moral elements
7. User adds any special requests
8. User selects target audience
9. User is directed to payment page
10. After payment, system processes:
    - AI story generation
    - Audio narration creation
    - Storage of assets
11. User is notified when story is ready
12. User can view, listen to, and download the story
13. User can access all stories from their dashboard

API Integration Points

OpenRouter AI API

- Endpoint for story generation
- Parameters include prompt template, character information, and setting details
- Handles response processing and error handling

Text-to-Speech API

- Converts generated story text to audio
- Options for voice selection based on character and target audience
- Audio file processing and storage

Stripe API

- Payment processing

- Webhook handling for payment status updates

Storage API (Cloudflare R2)

- Storage of audio files
- Generation of temporary URLs for audio streaming

Technical Requirements

Database

- SQLite database configuration in Laravel
- Migrations for all required tables
- Efficient indexing for common queries
- JSON storage for flexible metadata
- Regular backups of SQLite database file

Backend

- RESTful controllers for all main functions
- Service classes for external API integrations
- Job queues for async processing (AI generation, audio conversion)
- Proper error handling and logging

Frontend

- Mobile-first approach with progressive enhancement for larger screens
- Touch-optimized controls and interactions
- Step-by-step form interface similar to StoryAI
- Immersive book interface with page turns and visual enhancements
- Typography optimized for reading on mobile devices
- Persistent audio player with background playback capabilities
- Prominent download buttons for PDF and MP3 formats
- Loading states and progress indicators
- Touch-friendly navigation
- Proper handling of device orientation changes

Security

- Authentication and authorization
- Secure API key storage
- CSRF protection
- Input validation
- Proper file permissions for SQLite database

Performance

- Caching for common resources
- Efficient database queries
- Asset optimization
- Consider read-only connection for frequently accessed content

Deployment Considerations

- SQLite database file permissions and location
- Database file backup strategy
- Connection handling for concurrent users

This PRD and ERD provide a comprehensive blueprint for developing Make Story Time 2.0, leveraging modern technologies with SQLite as the database of choice for simplicity and ease of deployment.