Updated Project Documentation

File Name

- **Name**: `Project Documentation for _Song of the Day_ Application.md`
- **Location**: `C:\Users\Ntokozo\song-of-the-day\`

Complete Content

Below is the fully regenerated and updated documentation, reflecting the successful Netlify deployment, the development journey, and the full tech stack. Replace the existing content with this or create a new file if it doesn't exist.

markdown

```
# Project Documentation for _Song of the Day_ Application

## Overview
The "Song of the Day" application is a React-based web app designed to display a random song each day, featuring artwork, title, artist, album, and release year. Styled with a Spotify-like UI using Tailwind CSS, the app integrates with the Deezer API to fetch real-time song data. It offers a seamless user experience with a "New Song" button to refresh content, making it ideal for music enthusiasts. The project was developed as a personal learning exercise and successfully deployed on Netlify.

## Project Details
- **Author**: Ntokozo Ntshangase
- **GitHub Repository**: [https://github.com/tkntsh/song-of-the-day]
- **Live Deployment**: [https://tkntsh-song-of-the-day.netlify.app/]
- **Development Start**: October 21, 2025
- **Deployment Date**: October 22, 2025
- **Version**: 0.0.0

## Features
- Displays a random song with artwork, title, artist, album, and release year.
- "New Song" button to fetch and display a new random song.
- Responsive design optimized for desktop and mobile devices.
- Real-time data integration with the Deezer API.

## Tech Stack
See the "Full Tech Stack" section below for a detailed breakdown.

## Development Process
- **Initialization**: Started on October 21, 2025, using Vite to scaffold the React app and initialize a Git repository.
- **Styling**: Implemented Tailwind CSS for a modern, Spotify-inspired user interface with responsive layouts.
- **API Integration**: Integrated the Deezer API, initially proxied via Vite's development server, with Netlify redirects handling CORS in production.
- **Deployment Journey**:
```

  - **Heroku**: Initial attempt using Node.js and NGINX buildpacks encountered issues (e.g., H14 errors, deprecated static buildpack), leading to abandonment due to complexity.
  - **Vercel**: Tried with `vercel.json` rewrites but faced CLI errors (e.g., 404 NOT_FOUND) and configuration challenges.
  - **Netlify**: Successfully deployed on October 22, 2025, using `netlify.toml` for build and redirect configurations, resolving CORS and deployment hurdles.
- **Challenges Overcome**: Debugged deployment logs, adjusted configurations iteratively, and resolved CORS with Netlify's redirect rules.

## Installation and Setup
1. Clone the repository:

```
git clone https://github.com/tkntsh/song-of-the-day.git cd song-of-the-day
```

2. Install dependencies:

```
npm install
```

3. Run locally:

```
npm run dev
```

- Access at `http://localhost:5173` (default Vite port).

## Deployment Instructions
- **Platform**: Netlify
- **Steps**:
  1. Log in to netlify.com.
  2. Click "New site from Git" → Select "GitHub" → Authorize Netlify.
  3. Choose the `tkntsh/song-of-the-day` repository.
  4. Configure build settings:
  - Branch to deploy: `main`
  - Build command: `npm run build`
  - Publish directory: `dist`
  5. Add the `netlify.toml` configuration file (see below) to the project root.
  6. Click "Deploy site" and note the live URL provided.
- **Configuration File**: `netlify.toml`

~~~~
  ```toml
  [build]
 command = "npm run build"
 publish = "dist"
 environment = { NODE_VERSION = "18" }

  [[redirects]]
 from = "/api/*"
 to = "https://api.deezer.com/:splat"
 status = 200
 force = true
~~~~

- **Explanation**: The `[build]` section defines the build process, while `[[redirects]]` proxies `/api/search` to the Deezer API, bypassing CORS.

Future Improvements

- Implement audio preview functionality using Deezer’s streaming API.
- Add a daily song lock feature to limit refreshes to once per day.
- Enhance the UI with animations or a dark mode toggle.
- Incorporate error handling for API failures or network issues.

Credits

- **Framework**: Vite, React
- **API**: Deezer API
- **Styling**: Tailwind CSS
- **Deployment**: Netlify

Full Tech Stack

Frontend

- **React**: v18.2.0
    - JavaScript library for building dynamic user interfaces, powering the song card component.
- **Vite**: v4.0.0
    - Modern build tool and development server, enabling fast development and optimized production builds.
- **Tailwind CSS**: v3.0.0
    - Utility-first CSS framework for rapid, responsive styling with a Spotify-like aesthetic.

Styling

- **PostCSS**: v8.4.0
    - Processes CSS with Tailwind, ensuring compatibility and optimization.
- **Autoprefixer**: v10.4.0
    - Adds vendor prefixes to CSS for broader browser support.

Development Tools

- **@vitejs/plugin-react**: v4.0.0
    - Vite plugin for React integration and hot module replacement.
- **@types/react**: v18.2.0
    - TypeScript type definitions for React (optional, included for potential TypeScript use).
- **@types/react-dom**: v18.2.0

    - TypeScript type definitions for React DOM.

API and Data

- **Deezer API**
    - RESTful API providing song metadata (artwork, title, artist, album, year), accessed via Netlify redirects.

Version Control

- **Git**: v2.x (system default)
    - Source control for tracking changes and collaboration.
- **GitHub**: Hosts the repository at `https://github.com/tkntsh/song-of-the-day`.

Deployment

- **Netlify**
    - Hosting platform for the static site, with automatic builds from `dist/` and CORS handling via `netlify.toml` redirects.
- **Node.js**: v18.x
    - Runtime environment for building the app locally and on Netlify.

Testing and Debugging

- **Browser Dev Tools**: Used for network monitoring and debugging API requests.
- **Netlify Dashboard**: Provides deployment logs and site management tools.

Contact

- **Email**: ntokntshangase@gmail.com
- **GitHub**: tkntsh

- **Update Process**:
  - Open or create `Project Documentation for _Song of the Day_ Application.md` in a text editor (e.g., Notepad or VS Code).
  - Copy and paste the above content.
  - Save the file in `C:\Users\Ntokozo\song-of-the-day\`.
  - Commit and push to GitHub:
    ```
    ```

```
git add "Project Documentation for _Song of the Day_ Application.md"
git commit -m "Add complete project documentation with Netlify deployment and tech stack"
git push origin main
```