**Project Proposal: “MoodReads”**

**Overview**

This project attempts to enhance the experience of reading fiction by recommending music that matches the emotional tone of each book. Users will be able to search or browse books and get playlist suggestions that complement the mood, helping them feel more immersed in the story. As someone who enjoys fiction, I’ve found that the right music can make scenes more vivid and emotionally powerful. With this project I aim to bring those two experiences together in a simple, engaging way.

**Target Audience**

This web application is aimed at readers who want to enhance their reading experience or get immersed in the book they’re reading. For example:

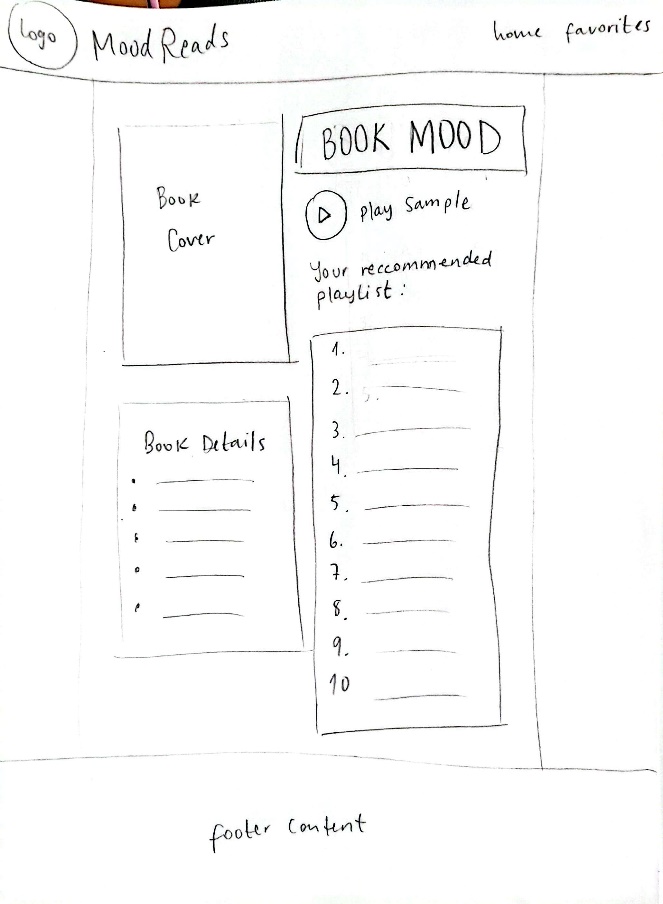
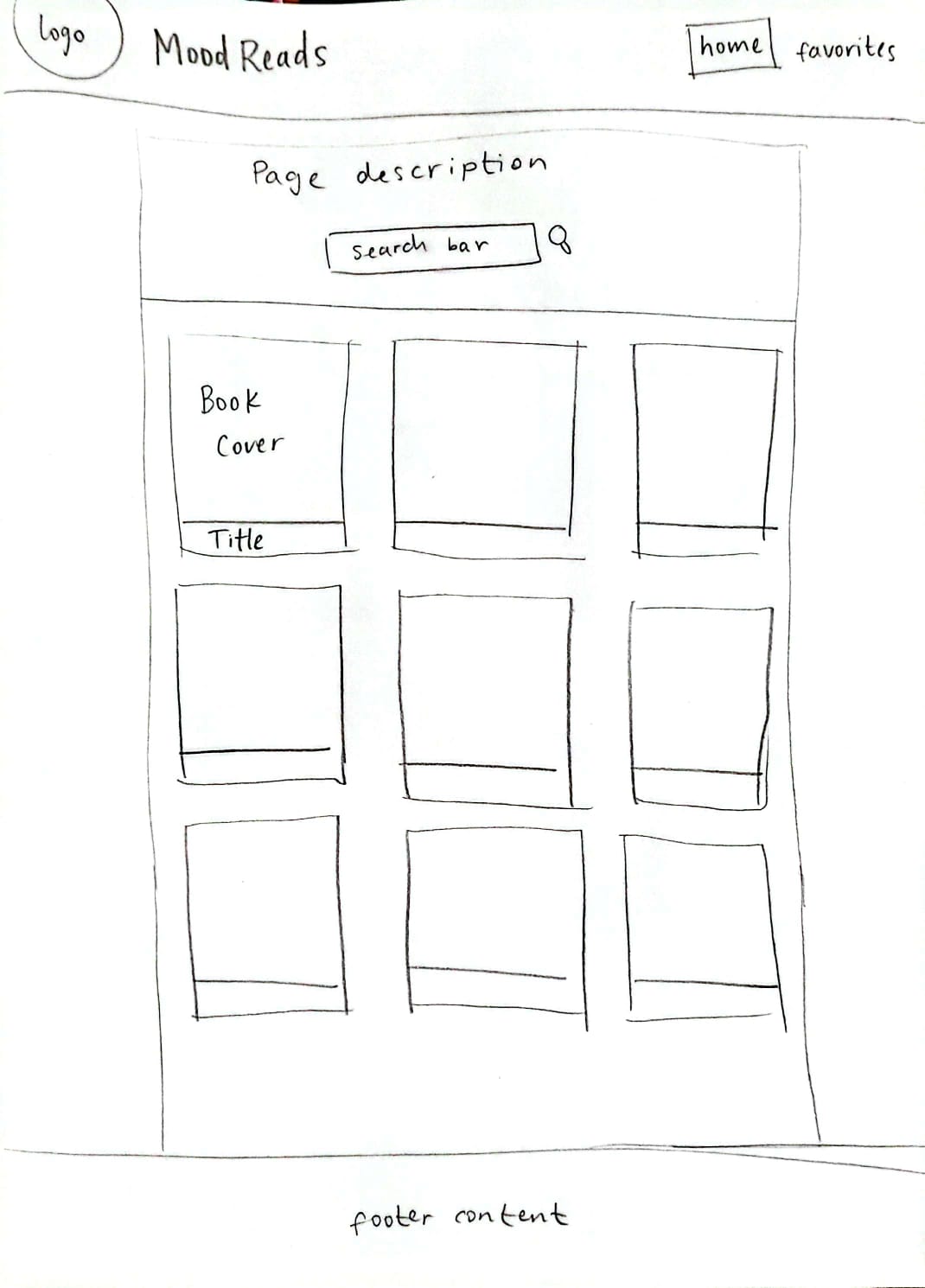
* Young adults and adults who read for leisure and enjoy customizing their environment.
* Students or book club members who want to enhance their focus or emotional connection with a book.
* Music lovers who enjoy discovering new playlists according to themes or moods.

**Major Functions**

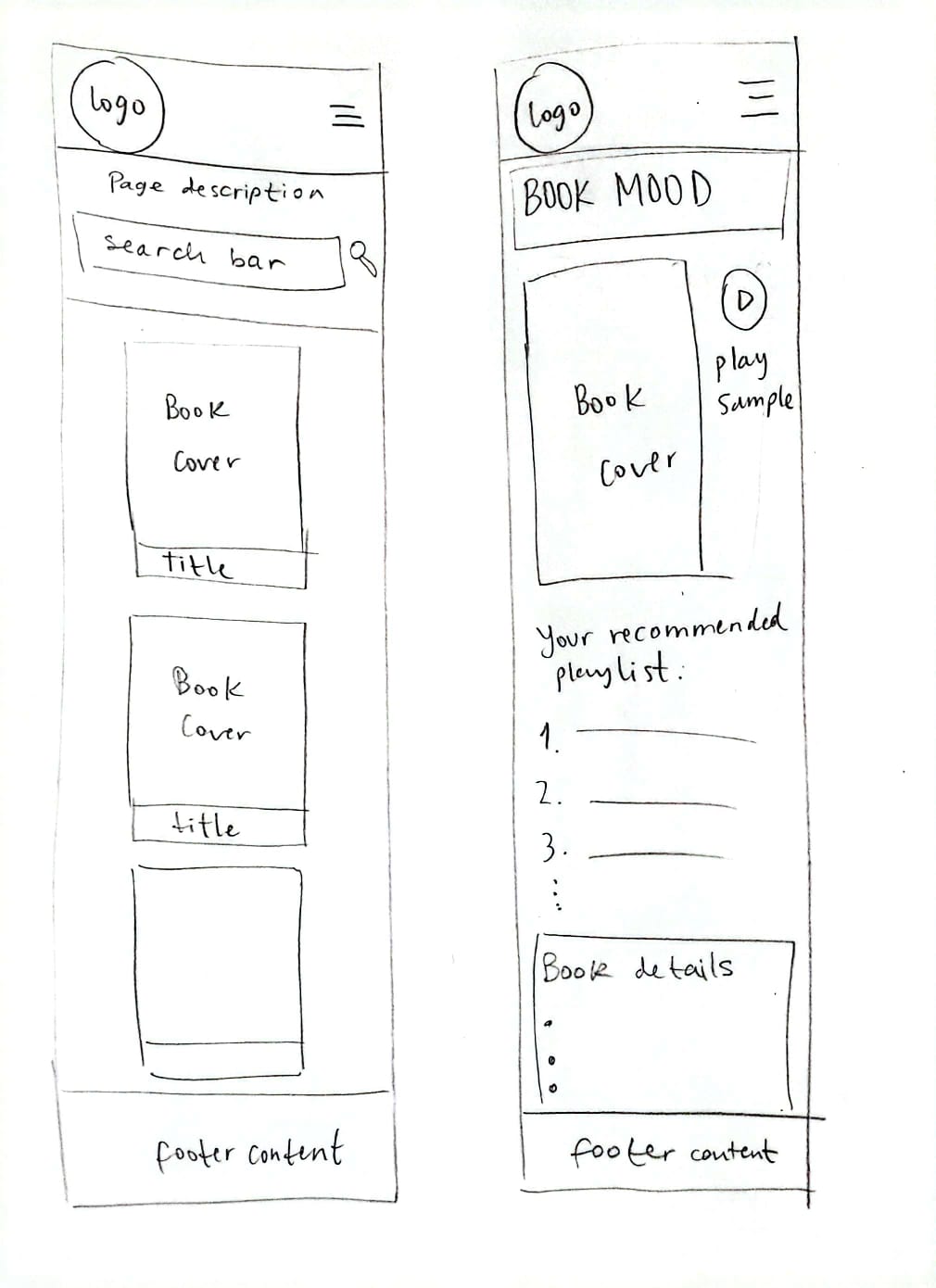
1. **Browse Popular Fiction Books:** When the page loads, the app automatically searches for a general term like "bestselling fiction" to display a selection of popular books. This provides users with content to explore immediately, even before they search for anything.
2. **Search for a Book by Title or Author:** Users can enter keywords to search for specific books. The app fetches and displays relevant results including cover images and basic information.
3. **Display Detailed Book Information:** Clicking on a book reveals more details such as the full title, author, description, average rating, and a larger book cover.
4. **Analyze Book Description for Mood:** The app sends the book's description text to the Twinword Emotion Analysis API to determine the main emotion or mood.
5. **Display Playlist Recommendation Based on Mood:** Based on the emotion detected, the app embeds a Spotify playlist that matches the mood of the book.
6. **Play a Sample Track Preview:** Users can listen to a short 30-second sample of a song from the playlist when clicking on a book.
7. **Save Favorite Book + Playlist Combos:** Users can bookmark favorite book and playlist pairings to localStorage for quick access in a dedicated “Favorites” section.
8. **Interactive UI Without Reloads:** The app loads all content dynamically using JavaScript, allowing for seamless interaction without full page reloads.
9. **Responsive and Accessible Design:** The user interface is designed to work well on both mobile and desktop devices, and includes accessibility features like alt text and keyboard navigation.

**Wireframes**

Home Page Book Page

****

Mobile Home Page Mobile Book Page

****

**External Data**

1. **Google Books API**  
   This API will be used to search for fiction books and retrieve details such as:

* Book title
* Author
* Cover image
* Description
* Average rating
* Categories
* Page count
* Publisher
* Language

1. **Twinword Emotion Analysis API**

This API will analyze the book's description text and return the dominant emotional tone, which will be used to recommend a matching music playlist.

**Data Storage**

To allow users to save and revisit favorite Book + Playlist pairings, the app will store at least the following three properties in localStorage:

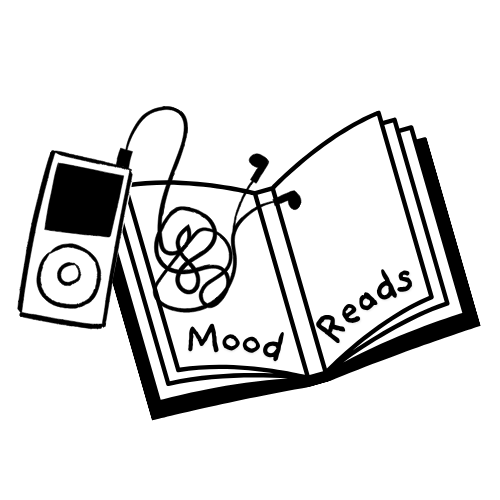
1. **Book Title:** To identify the saved book.
2. **Book Cover URL:** To visually display the book in a “Favorites” section.
3. **Associated Mood**: To remember the emotion that generated the playlist recommendation.

**Module List**

1. **Book Search Module:** Lets users search for books by title or keyword. Sends the request to the Google Books API and handles the results.
2. **Book Display Module:** Renders book cards on the page using data from the API.
3. **Emotion Detection Module:** Sends the book’s description to the Twinword Emotion Analysis API. Gets the emotion result.
4. **Playlist Recommendation Module:** Correlates each emotion to a pre-selected Spotify playlist. Embeds the playlist into the book card or popup using Spotify’s iframe.
5. **Favorites Module:** Lets users save books to localStorage as favorites. Stores at least 3 properties (title, author, emotion).
6. **Favorites Display Module:** Retrieves favorites from localStorage and displays them on a separate section or tab. Allows unfavoriting/removing items.
7. **Initial Browse Module:** Automatically loads a preset category like "fiction" or “bestsellers” when the page first loads so the site is populated from the start.

**Graphic Identity**

* **Color scheme:**
  + Tiffany blue #8EC2BD: Primary Color for main content
  + Payne’s Grey #49616D: Secondary Color for secondary content
  + Dutch White #FCEAC3: Accent Color for small accents/decorations
  + Oxford Blue #03001D: Dark Color for dark backgrounds or text color
* **Typography:** 
  + Schoolbell: For main headings
  + Lato: For everything else
* **App Logo:** The logo is a book with a music device with wired earbuds on top, this signifies the experience of immersing yourself in a book while listening to music, the name of the app “Mood Reads” is also included. Color may be added later.

****

* **UI:** The UI will be minimalistic and simple since the book covers displayed in the homepage will bring a lot of color and different styles to the site.

**Timeline**

**Week 1: Core Setup & API Integration**

* Set up project layout and connect to Google Books and Twinword Emotion Analysis APIs.
* Display search results and analyze book descriptions to determine emotional tone.
* Start mapping emotions to curated Spotify playlist embeds.

**Week 2: Interactivity & Styling**

* Implement favorites feature with localStorage for saving and retrieving books.
* Display default book category on page load and embed corresponding playlists.
* Begin adding CSS styling and animations for smoother UX.

**Week 3: Polish & Final Touches**

* Refine UI, complete responsive design, and polish animations.
* Test all features thoroughly and ensure dynamic JSON usage is solid.
* Prepare presentation materials and ensure code is clean and commented.

**Project Planning**

<https://trello.com/b/iL0sEirQ/book-playlist-project>

**Challenges**

1. **Figuring Out the Mood of Each Book:** It might be hard to get the right emotion from a book description because the text can be vague or have more than one feeling.
2. **Showing the Right Playlist for the Mood:** It could be tricky to make sure each mood connects to a good playlist and that the music shows up correctly on the page.
3. **Making the App Work Smoothly with the APIs:** Since the app has to get information from two APIs at the same time, it might cause delays or make the site feel slow if it’s not handled well.