



Prepared for FSD07 - WD2

Prepared by
Arjun Dhakal
Mihai Coanda
Mackenzie Sawyer

June 16, 2023

Project Overview

Building a crowdsourcing platform similar to the KickStarter investment model, focusing on fundrasing for different artistic ventures: music, indie film, art instalations, photography projects etc.

Functional Requirements

- user registration
- project creation
- contribution submission
- project browsing
- project sorting

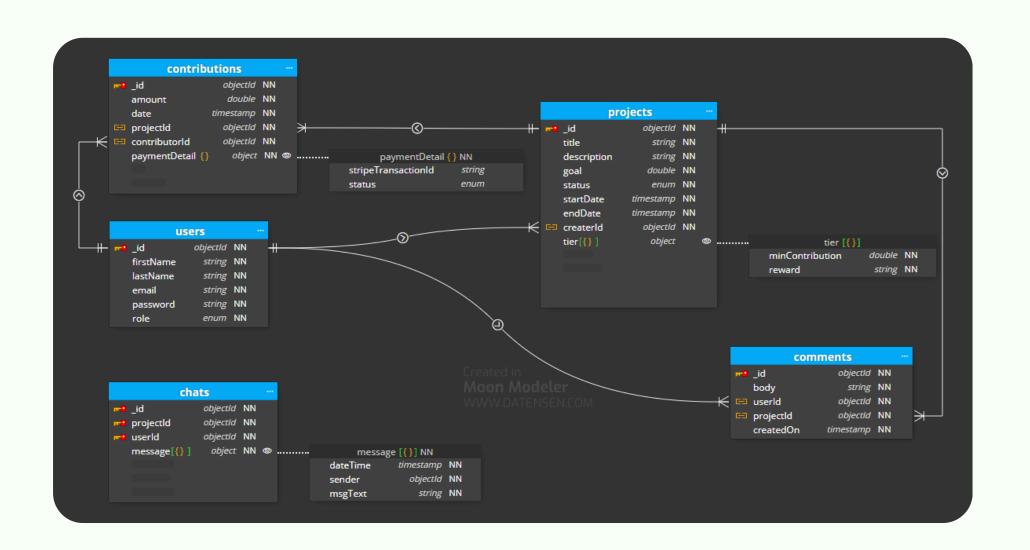
Non-Functional Requirements

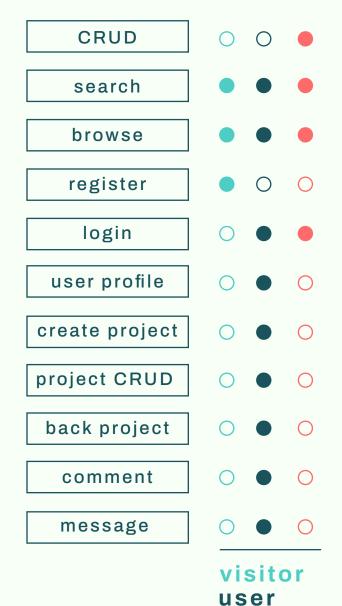
- fast and responsive page load times
- secure payment processing
- intuitive user interface
- support for many concurrent users

Technologies used

- - React
- - Next.js
- - Node package manager
- - Bootstrap
- - Firebase
- - Stripe API
- - HTML
- - Javascript
- - CSS

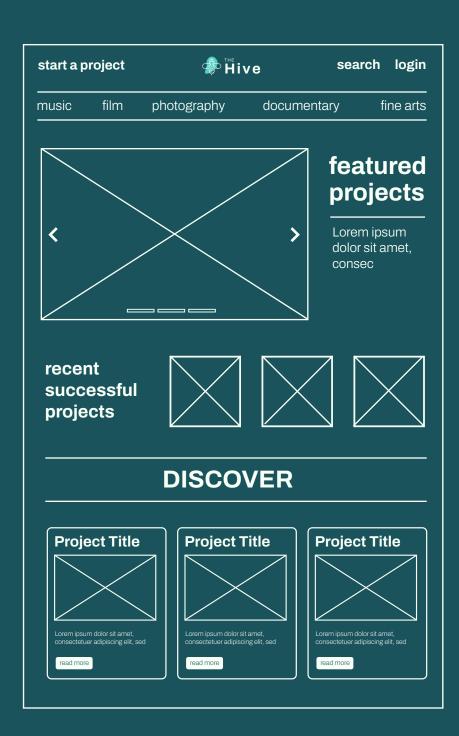
Database Design & Use Case

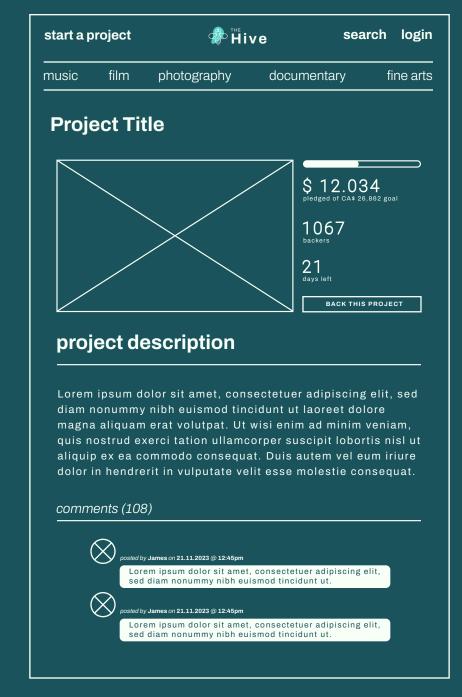




admin

Wire Frames Examples





As highlighted in our previous projects, wireframes play an essantial role in getting the project off the ground, by providing a clear roadmap.

Furthermore they help identify and prioritize tasks.





MEGAPUFF: Origami Solar Power Bank And

emergency situations where maximum visibility is crucial. Greater POWER + LIGHT. Say goodbye to a

read more \rightarrow

successfully funded projects







Discover & Share



MEGAPUFF: Origami Solar Power Bank and Bigger + CHARGE YOUR PHONE! Now with Phone



Story Say goodbye to short nights and embrace the



Story Hi, we are Present - a brand dedicated to enabling a



NOT just another Bach recording: a community

Let's connect: 🙌 🧿 💆





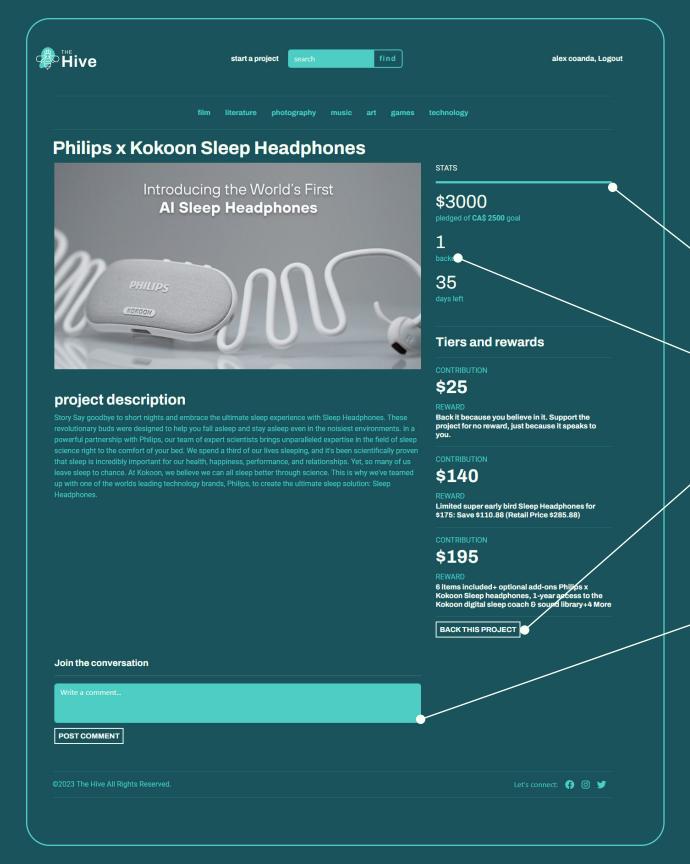
Site Features

personalized menu

category selection

dynamic slideshow

reactive UI



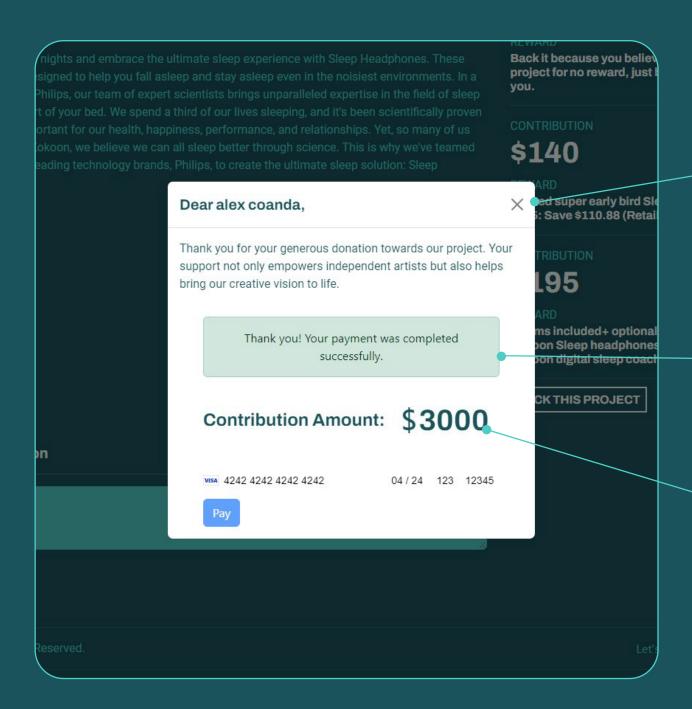
Site Features

dynamic visualisation

live reload fields

selective UI rendering (user vs. visitor)

comments section



Site Features

personalized payment pop-up

payment success / error

minimum contribution validation

Challenges

- time management, prioritizing tasks and functionalities while allowing time to learn new skills
- solution: frequent team meetings, support from teammates + teacher, sharing learning resources
- routing API server side functions ex: Stripe API needed pages/api/create-payment-intent.js
- passing a prop from parent component to child component & modal
- ex. projectDetails.js passes 'projectId' to CheckoutForm.js, StripeModal.js & Comments.js

Challenges

- submit "contributions" & "comments" collections to parent collection, "projects"
- link with user UID & projectId

- Javascript conditional CSS & dynamic CSS

- Callback functions
- ex. when pulling records from database they would not populate field immediately, solution: useState

Challenges

- Firebase database
- no-cost limit reached during development (useState code firing automatically, solution: useEffect)
- learning Firebase (not a SQL style database change our approach to database design because there are no foreign keys linking tables)
- difficult to categorize by fully funded projects (successfully funded projects), loop by n^2, inefficient way to sort because each project have to loop through all contributions
- Flaw in database design, we need alternative solution, update 'status'
 of project to completed when total 'contributions' >= to 'goal' amount,
 add function that calculates this status

Thankal // Mihai Coanda // Mackenzie Sawyer