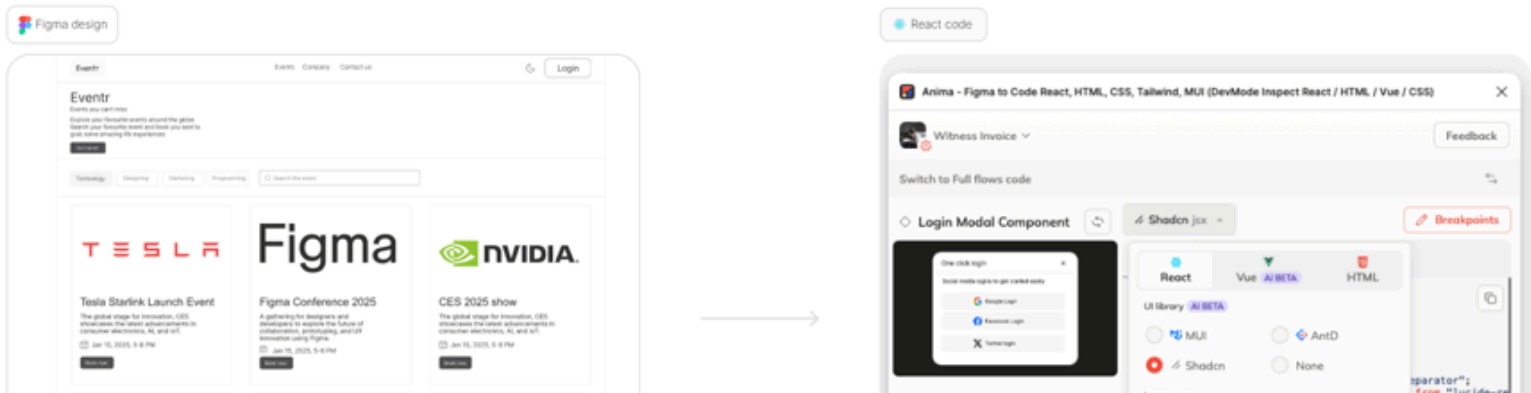


# Figma design → React Code

Anima figma plugin + AI code editor  
powered by 



 **shrey vijayvargiya**  
Posted on Mar 19

 1  1  1  1  1

## Figma to React instantly: Introduction Anima app

[#webdev](#) [#ai](#) [#react](#) [#news](#)

### Figma design → React/Vue/HTML code instantly

Hello people,

Welcome to the new blog. Today we will be discussing a cool product that converts figma design into code, and will make code our first app page using the same.

Taking some time off because I want to thank this blog sponsor and supporter, [Anima](#): the company behind the AI figma design to code plugin and the AI code editor.

Anima [Figma plugin](#) converts Figma design to the code of your choice. Developers are React.js, Vue.js or Plain HTML working with shadcn or different UI libraries, one click and your figma design will be converted into code.

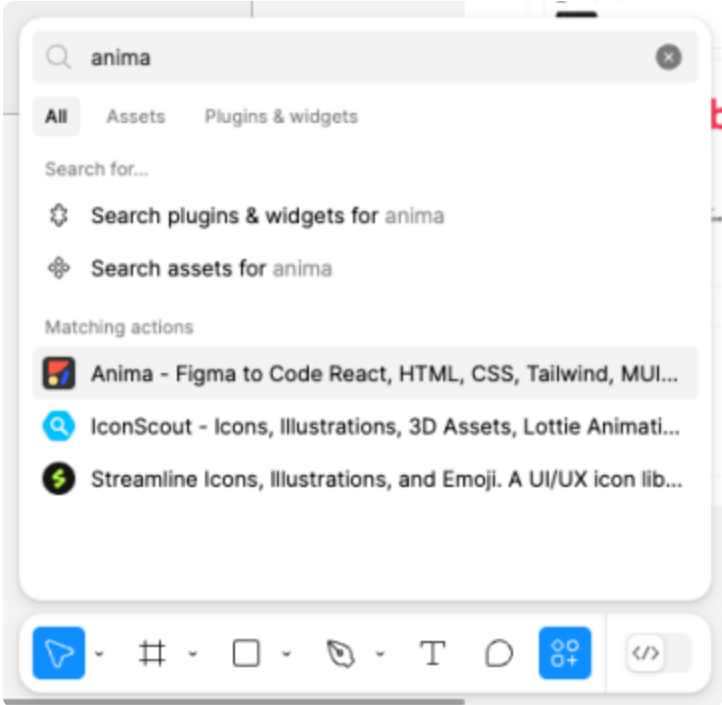
In this Blog!! (Table of Contents)

1. Installing the Anima figma plugin
2. Designing event registration app page (real-world example)
3. Figma plugin details(AI customisation, Code customisation)
4. Designing -> Developing a Registration app page using the Anima figma plugin using a login form
5. Introduction to Anima Playground (AI Coding Assistant)
6. Implementing Registration Form
7. Search functionalities added using fuse.js (Using open-source package in AI chatbot)
8. Contact Us page (Testing image uploaded feature)
9. Company page generated using AI (Testing AI suggestions)
10. Handling errors in AI coding assistant
11. Download the code repository
12. Running the code repository locally
13. Final results
14. Conclusion

Let’s get started !!

## Installing the Anima figma plugin

You need to install the Anima figma plugin in your figma app to convert your designs to code.



Quick steps to install the plugin

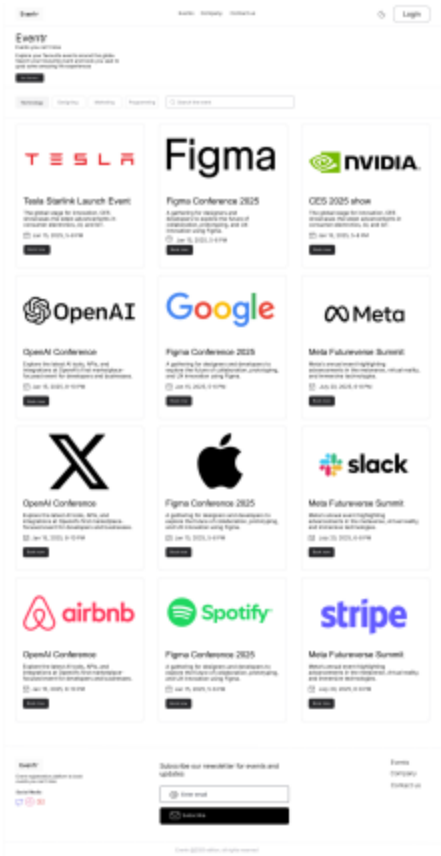
- In the figma app, navigate to the bottom fixed navigation tab or simply press CMD + P inside
- This will open the search bar as shown in the image above
- Search for the Anima plugin and click to INSTALL it
- You will need to log in/sign up on the [Anima app website](#) and to use the figma plugin.

Once done, you can use the Anima plugin directly inside figma.

## Designing Event Registration Page

We will be making a simple Event Registration App page and will convert it into an app using an Anima plugin

Here is the [figma file URL](#) to our Event Registration Page.



### Web Page Structure in 4 Points

**Header:** Displays the company brand name, includes navigation links to other pages.. Contains a "Login" button.

**Body Content:** Features an events list with filters (tabs) and a search bar, each event includes the name, description, date, and a "Book Now" button. Clicking this opens a modal with a booking form, A "Get Started" button provides information about the platform.

**Footer:** Showcases the brand name, includes a newsletter subscription form, social media links, and links to other company pages.

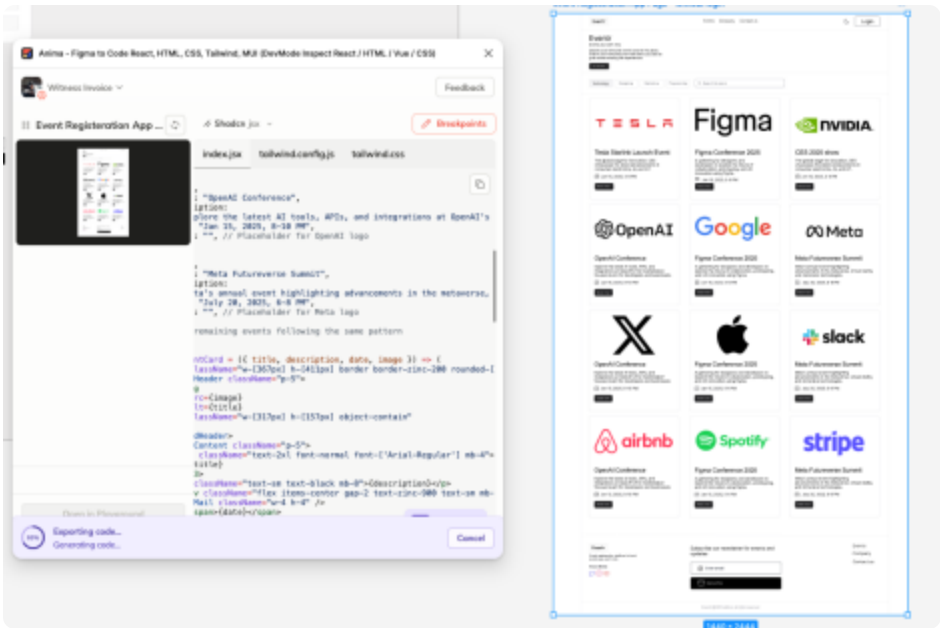
**User Interaction:** Smooth navigation and filtering with search and tabs, Booking form modal for seamless event seat reservations, Easy access to company info via footer links.

This is a very basic registration app, feel free to make changes according to your preference.

### Design to Code Using Plugin

Follow are the basic steps to convert your design to code

1. Open the Anima plugin using **CMD + P**
2. Search for the Anima plugin (install it if not present)
3. Authenticate within the plugin (it will redirect to the Anima website) 4. Create the design(feel free to use [anima figma designs](#))
4. Select the Figma frame that you want to convert into code
5. Further select the language, framework, and UI library in the plugin dropdown



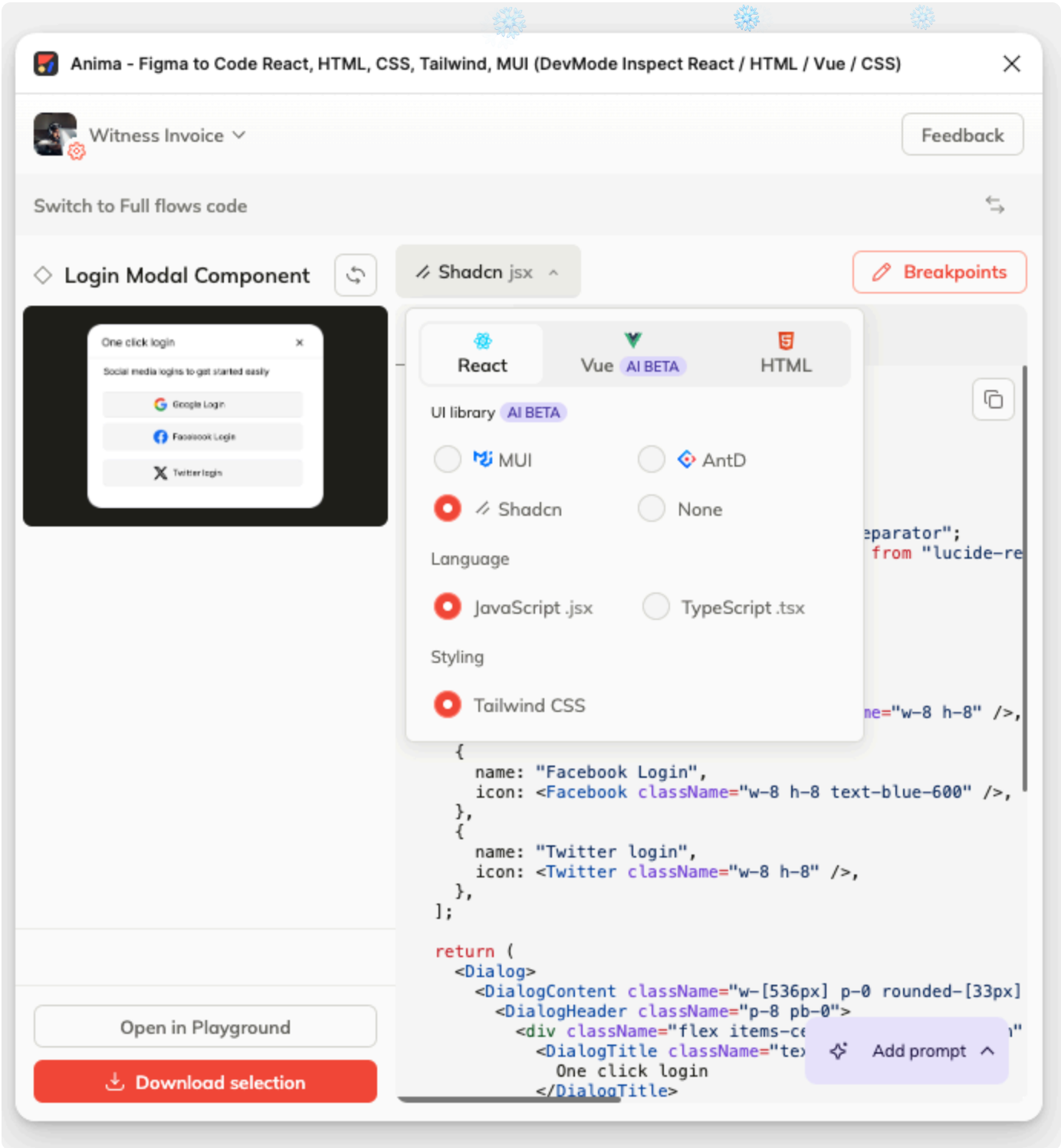
For the UI library, we have 3 options

- Ant design,
- Material UI,
- Shadcn

and 3 options for frameworks

- React
- Vue
- HTML

Options for styling are Tailwind CSS, and for language, we have 2 options, JavaScript and typescript.



Moving further, to using the playground for further development of our event registration page

## Introduction to Anima Dev Playground

The playground app by Anima runs the code repository in the browser to customise further using an AI chatbot.

In the figma plugin, once the frame is selected and code is generated by Anima AI, click on Open in Playground to redirect to the Anima AI dev playground.

You can use an Anima browser-based code editor called Playground that provides an AI chatbot to make changes in the code while typing in plain english, the same way we did with the figma plugin AI customisation chatbot.

The following tasks can be accomplished using Playground

- Customise code into Fledge apps using AI chatbot
- Preview the code changes in real-time
- Browser-based code editor with integrated AI chatbot
- Download the code repository
- Push the code repository to github
- Save the code repository under your account on Anima or on your codebase

Anima Dev Playground is scheduled to go live on February 10th, and for that day we can't wait to see your opinions/feedback and reactions.

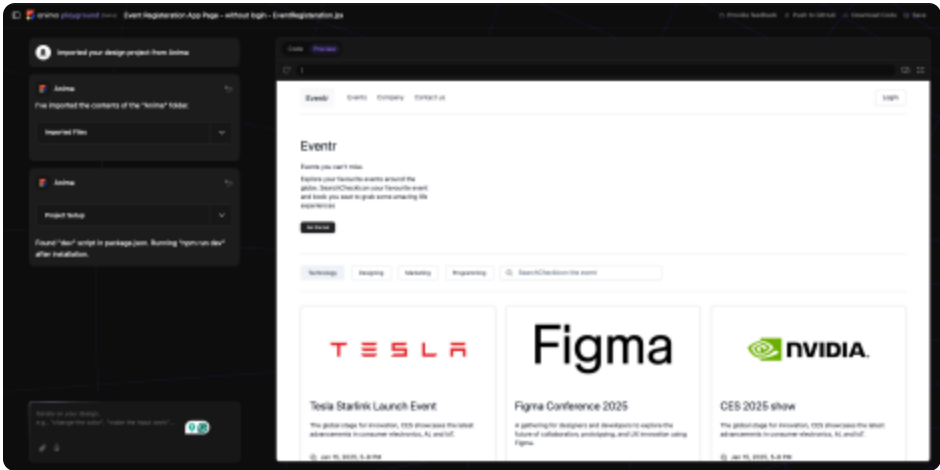
## Prompts Iterations in Anima Playground

Anima not only provides a figma plugin but it takes a step further with a browser-based AI assistant code editor, we will call it Dev Playground for this blog section (or probably in future.

Dev Playground is the AI coding assistant that runs on a browser whose job is to take prompts and generate code and apps in real-time.

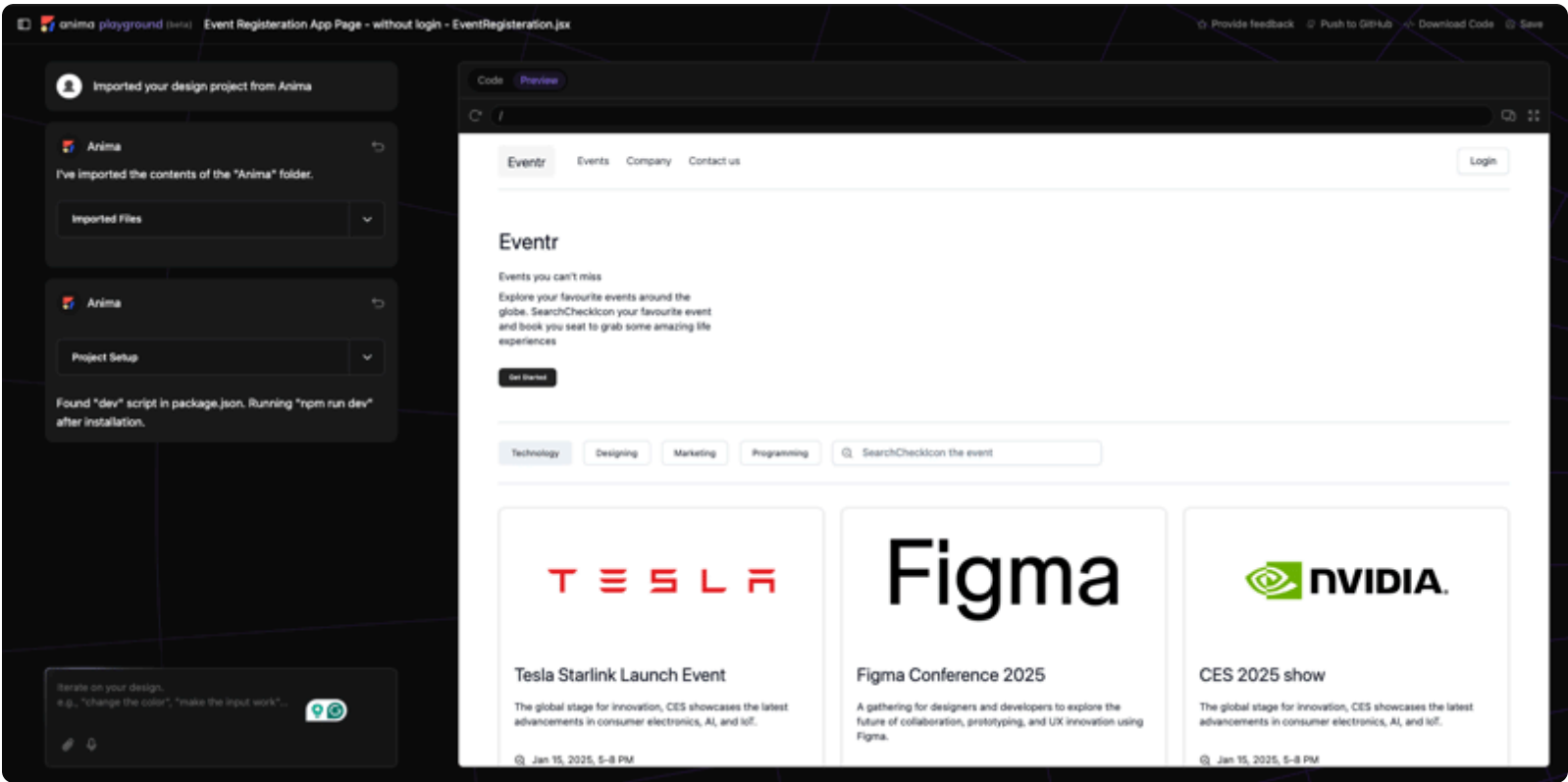
Following are the steps we will follow, or simply instruct our dev playground in the same order as mentioned

- Implement a login button in the header that opens and closes a modal on click. The modal should include an email/password login form and buttons for Google, Facebook, and Twitter authentication.
- Implement a 'Book Now' button that opens and closes a registration modal on click. The modal should contain event details and a form for user inputs like name, email, phone number, profession, a content consent checkbox, and a submit button.
- Complete that search input functionality by adding a search feature on the events list
- Complete the newsletter subscribe form as well by adding toast when the user adds an email, and click the subscribe button



3 core features the Anima playground provides

- The top navbar contains the Download, Save and Push code to github feature
- Playground provides a preview and code tab to see the results and code repository
- The playground left-side section (as shown in the image above) will take AI instructions via prompts and show previous chats and steps.

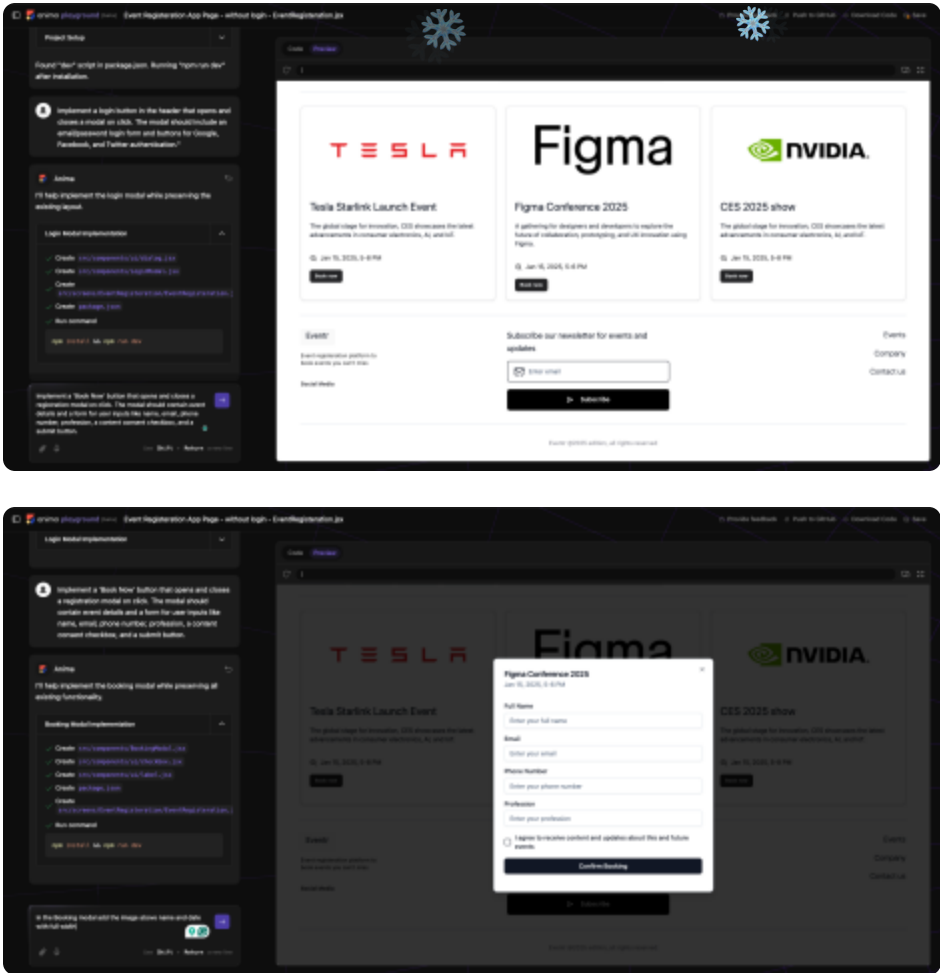


Moving ahead with further customization of the event registration page by adding a registration form, login modal, newsletter subscribe feature and search functionality.

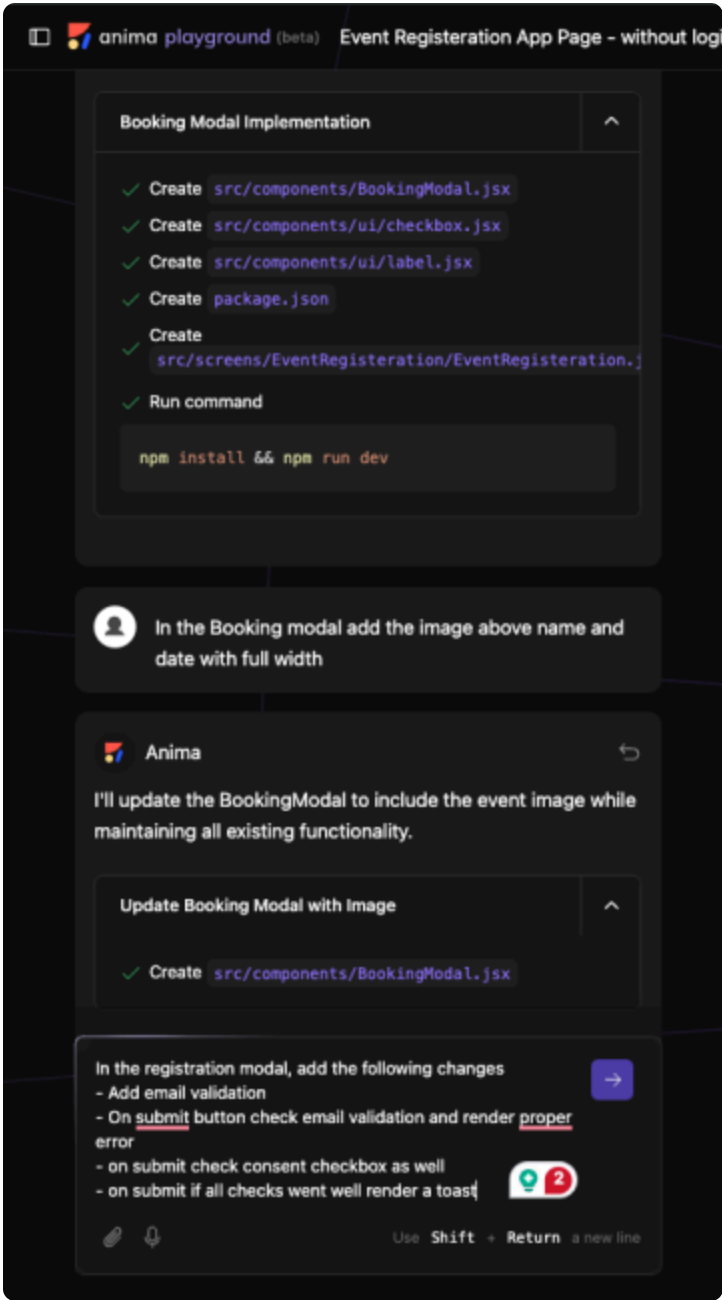
## Implementing the Registration Form

The following images depict the use case of the playground AI chatbot along with prompts to implement the registration form.





Lastly, we instruct AI to complete the registration form by taking email validation, consent checkbox confirmation and so on, and render a toast as well when a user submits the form.



Prompt to add registration modal along with registration form and handle the form submission with some toast. You can go further with AI assistant to make your app closer to full-fledged apps.

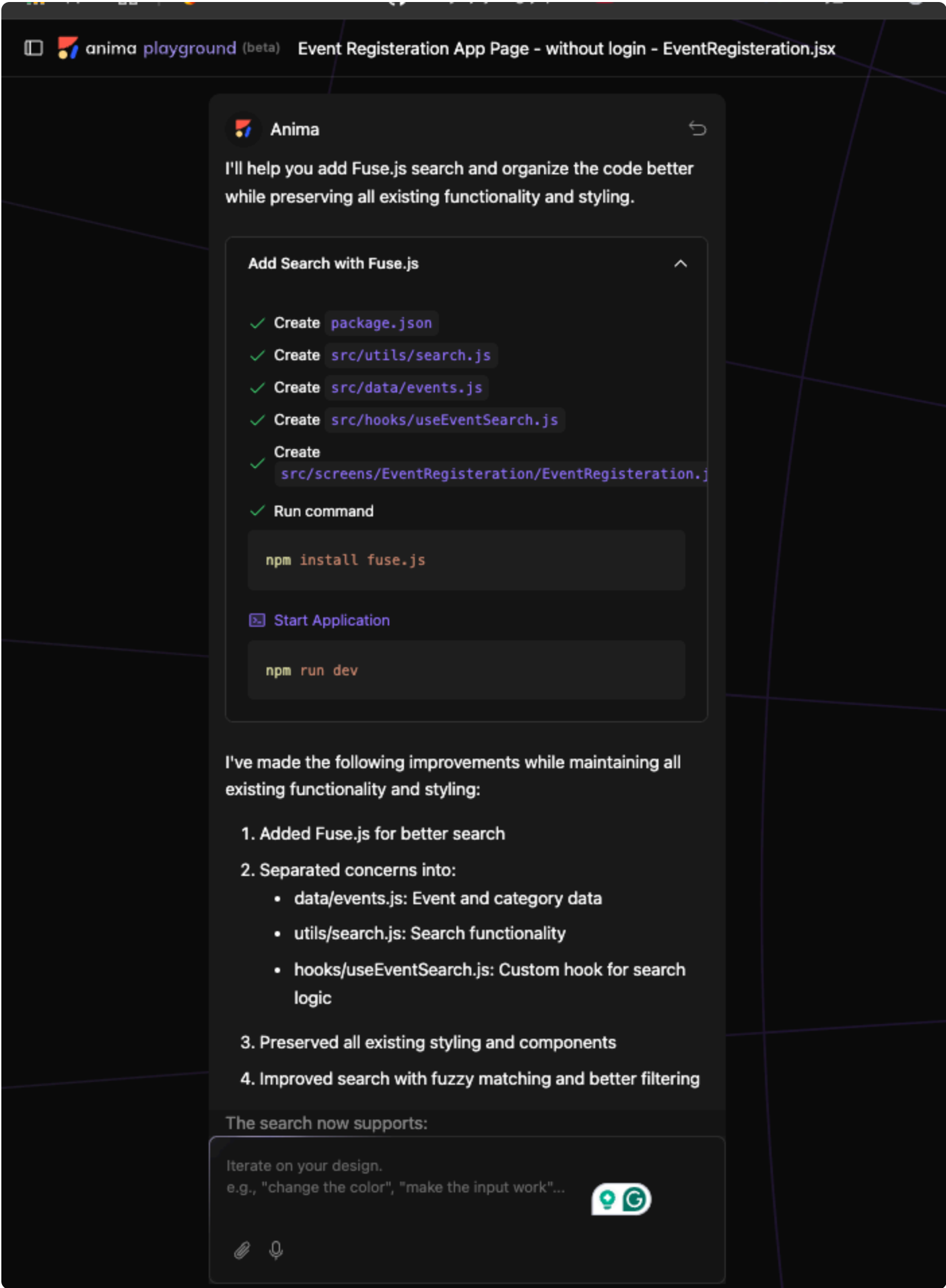
Complete the search functionality in the search input and show the searched result below

Anima figma plugin and AI coding assistant playground help to create basic figma design into code and develop full-stack apps using AI chatbot making life much easier for design developers.

## Creating Search Functionality using AI

Next, we will move to add basic search functionality using AI prompts by following the same process we did for the login and registration form, with only a difference this time we will instruct using other open-source libraries such as fuse.js to add search.

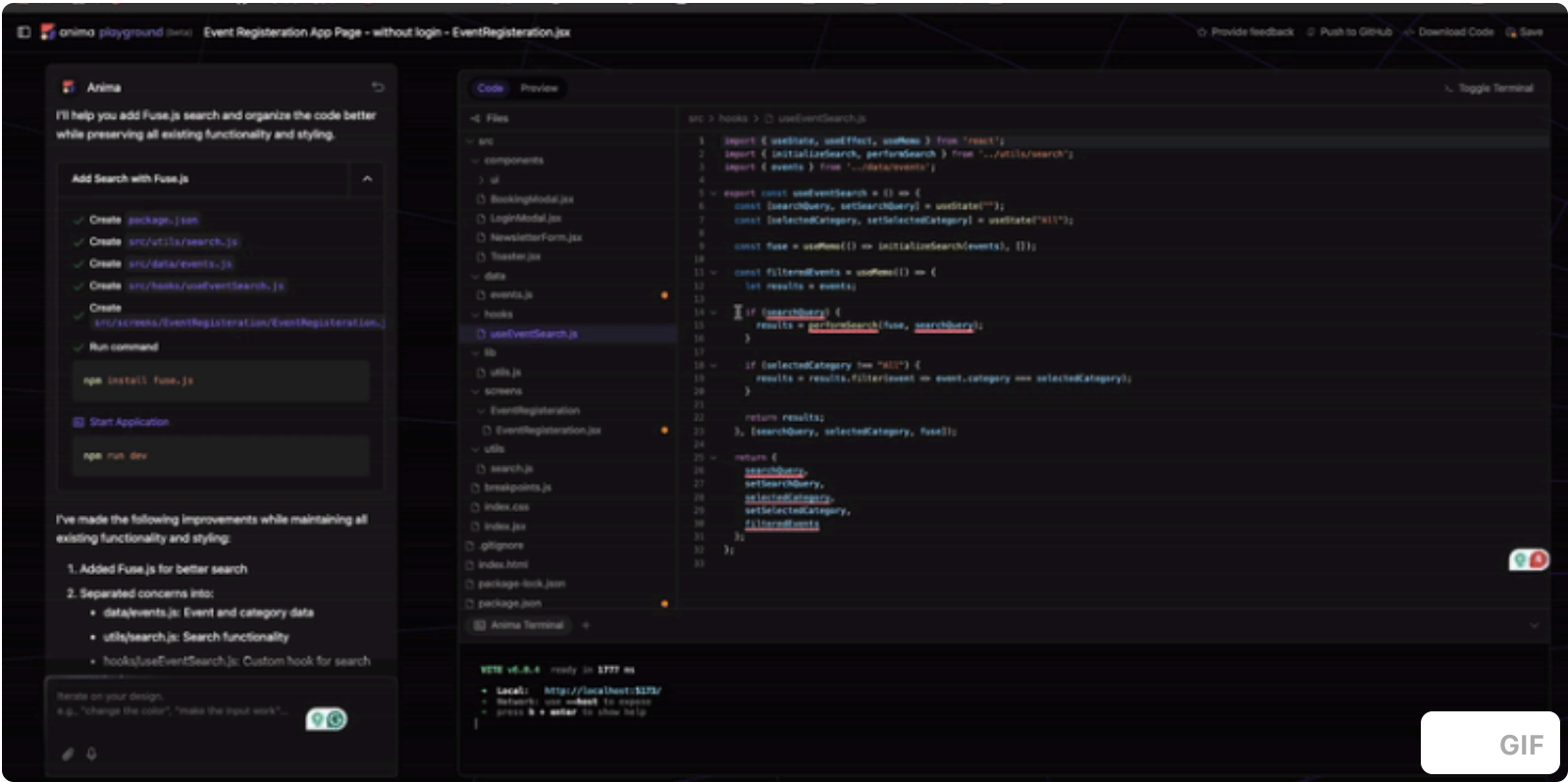
Prompt: Add basic search functionality on the tabs using the fuse.js npm package. Try to decouple the code and methods in a separate file.



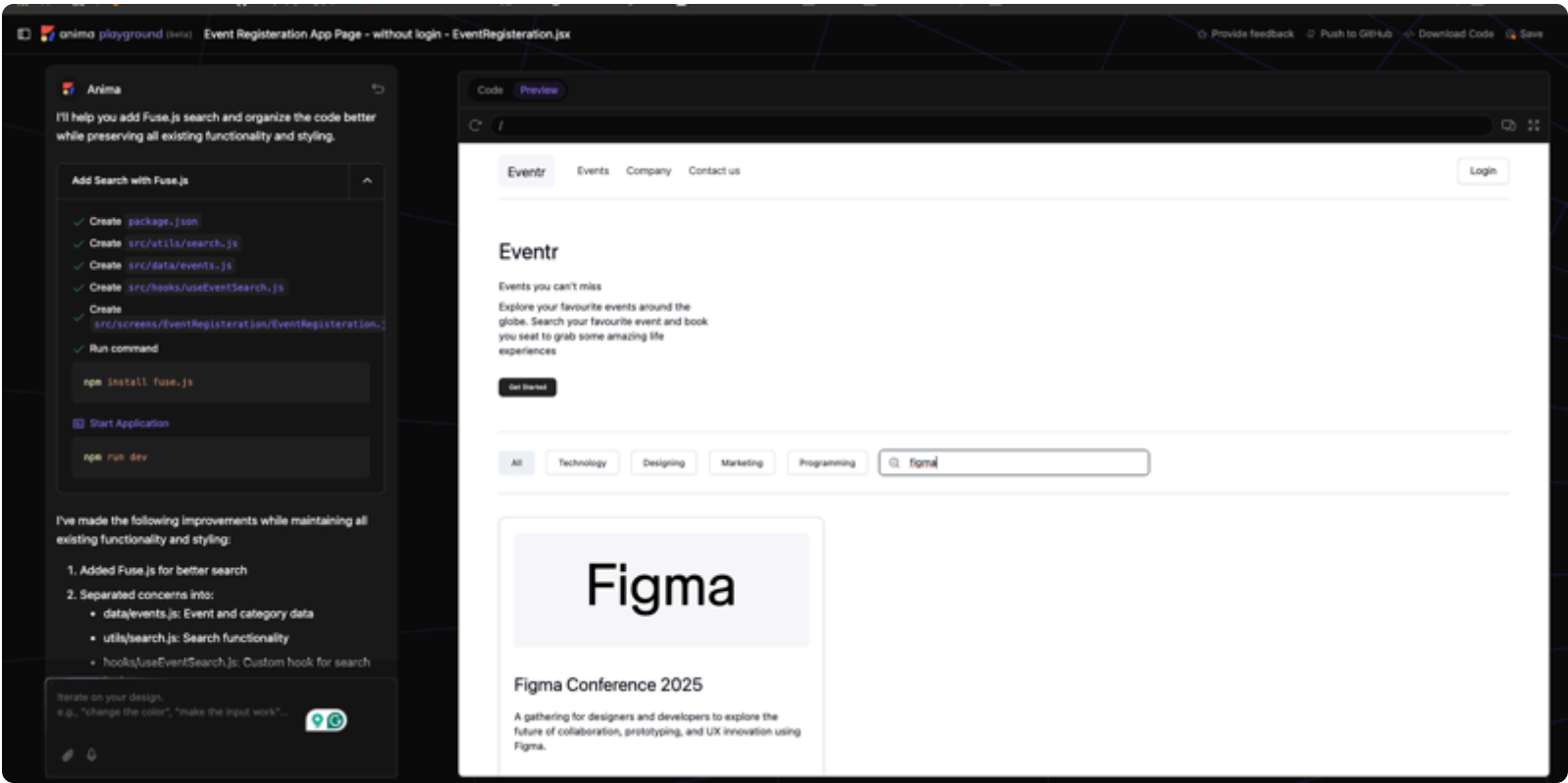
Since we have instructed the AI to decouple the code as well, it will certainly act as shown in the video below

As you can see, we have instructed the AI to handle decouple as well, and as expected, the playground installed the fuse.js package, then created the “utils” directory adding the fuse.js search method and importing the same to complete the search functionality.

All these tasks are completed instantly within minutes 😊 making it robust and comprehensive, helping to achieve the detailed desired output.



GIF



Anima AI Playground showing the app page search functionality installed using fuse.js

Moving ahead with adding one more tab button, one thing we deliberately missed in the design is the “All” label tab to show all events and the same needs to be done on our code repository, simply using Anima dev playground.

**Prompt:** In the above tab section, and add the label tab button with the tabs functionalities showing all the events under this tab.

So far we have the following features on the app page

- Navbar and banner with event details, branding
- User authentication modal containing the login form
- Events list along with tabs to search category
- Search input to search for events
- Event registration modal containing the registration form
- Footer displaying branding, pages, social media and newsletter subscribe form

Still, our app page needs more changes to move ahead with creating other pages, such as the other url pages, such as events, contact us etc.

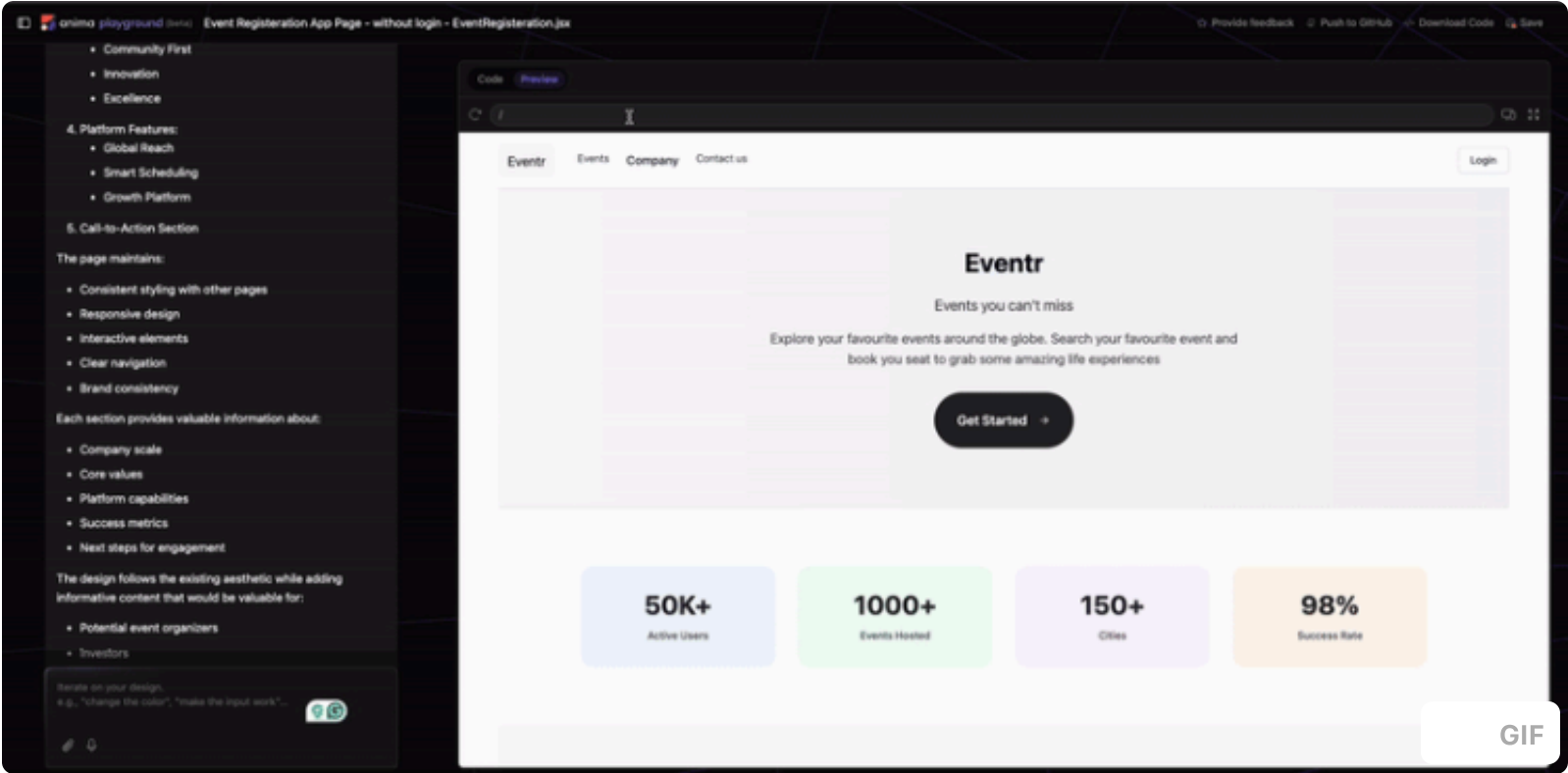
Feel free to adjust the choice of prompts and even the english to your needs. It depends on the developer's style of programming and the tech stack. For example, one can prefer other search libraries like Algolia to add search functionalities on the search input. In all those cases, totally feel free to instruct well to the dev playground AI chatbot.

Moving ahead with creating an “/events” page listing all the events in ascending order of the current date.



**Prompt:** Create an events page on /events route listing the existing events along with the book now button. The events page will have the same Navbar and Footer at the top and bottom of these events

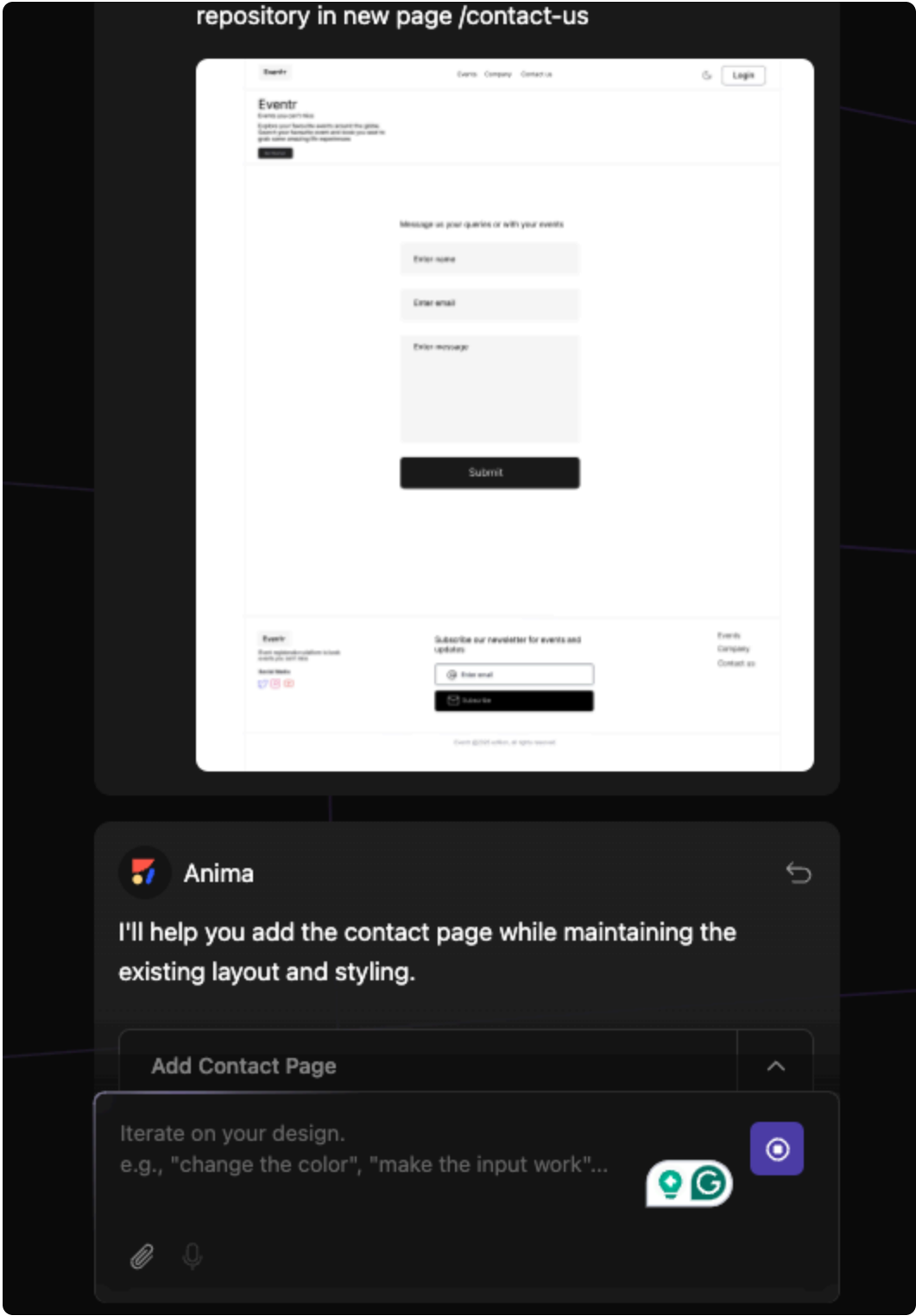
Video added below showcasing the events route page generating in the dev playground.



## Adding Contact Us page

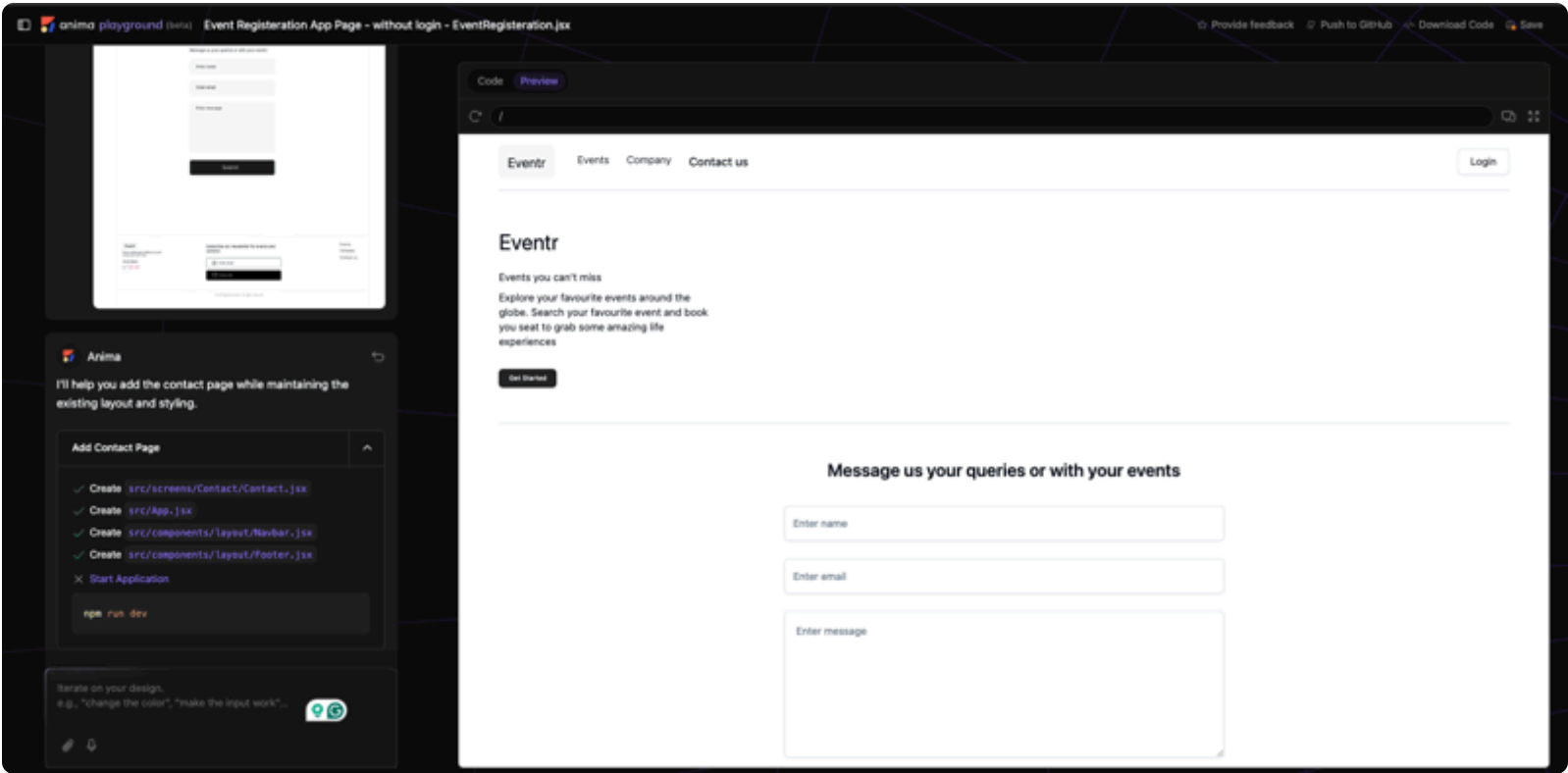
Next is to create a Contact Us page as we did for the events page.

This time we will follow another trick in between by instructing the AI in words, we will provide a figma downloaded file attached to the chatbot, followed by instructing the AI to create the same design on a new page with URL `"/contact-us"` (demo shown in the image below).



Anima Dev Playground takes care of handling code output according to the design file/image provided as an input (as shown in the image above), handling the app design changes in between development will save tonnes of time beforehand.

Every app development journey goes through design changes and website improvements and AI efficiently handles these issues.



## Creating Company page

Moving ahead with creating a final page is the company page.

This time we will ask something of our AI to suggest what we can add on the company page. Let's check how it goes with Anima AI in suggesting the company page design along with the content.

**Prompt:** Similar to the Contact Us page, add a company page on /company URL, but it doesn't need to have a contact form, so what else can we add, can you suggest a few more details?

This makes AI playgrounds more fun to use because AI creates rather new pages as per previous designs, just making things much faster in time.

AI output: AI created a company page including company details, geographic customer presence, why to choose Eventr as a platform and a Contact Us page. Within a few minutes, your new design page about the company details is ready(Check the demo video below)

Once this is over, we will give the final touches such as

- Handle responsiveness on every page of the website
- Complete the links by adding the URL
- Complete the Footer page social media by adding Twitter, Linkedin Instagram and other social media links
- Add more details to the event registration page
- Create a dedicated single page for each event along with book now button
- Handle SEO for each event page

Feel free to make changes accordingly by simply adding prompts to the AI.

## Connecting Database

The next step is to connect to the database and start collecting a few collections of data, such as user email of subscribers, user bookings and user authentication.

- Booking data
- Authentication data
- Subscriber data

We will be using [Firebase](#) for our event registration, and for the newcomers to this blog, Firebase provides a serverless database, authentication and storage and more functionality. What is

serverless, in simpler words, is the backend without your servers or collection of APIs.

As frontend developers, we don't need to write backend APIs for authentication, storage and database CRUD operations, those will be handled by most of the provided serverless databases and a few examples are Firebase, Supabase and Appwrite.

Moving ahead with Firebase, go to the Firebase console, create an account, followed by creating a project, you can use "event-registration-app"(for now) and get the API keys from the console in the project settings.

The image shown below is our event-registration-app Firebase console project API keys, please do not copy-paste them, it won't work in your project. One should love the same, just the values would be different, but the structure would be the same.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyANQzBD45ehjpoH4kH4L-x9q--RSnI9sfY",
  authDomain: "event-registration-app-a43d9.firebaseio.com",
  projectId: "event-registration-app-a43d9",
  storageBucket: "event-registration-app-a43d9.firebaseio.com",
  messagingSenderId: "1092280400765",
  appId: "1:1092280400765:web:7be6379de6ea9bc3782a14",
  measurementId: "G-QLLSM1BGQS"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

**Note:** This option uses the [modular JavaScript SDK](#), which provides reduced SDK size.

We will require the env keys and each key is named in a format that clearly/somehow dictates its purpose. For example, storageBucket is the storage bucket storing the files.

Then we will move to the dev playground and do the magic using prompts, and writing code using plain english.

**Prompt:** Write code to install the Firebase package, configure it for authentication and booking data collection, and set up environment keys from a .env file. Activate Firebase and ensure the configuration is ready to use

Few prompt variants examples,

- "Generate code to install and set up Firebase for authentication and booking data management. Include steps to load environment variables from a .env file and initialize Firebase."
- "Provide code to configure Firebase for user authentication and storing booking data. Use environment variables from a .env file for sensitive keys."
- "Create a script that installs Firebase, imports environment keys from a .env file, and configures Firebase for handling authentication and booking data."
- "Help me write a Firebase setup script for authentication and data storage. Include importing environment variables and initializing Firebase using a .env file."

The next step is to instruct AI to add a method to collect the booking data in the Firebase.

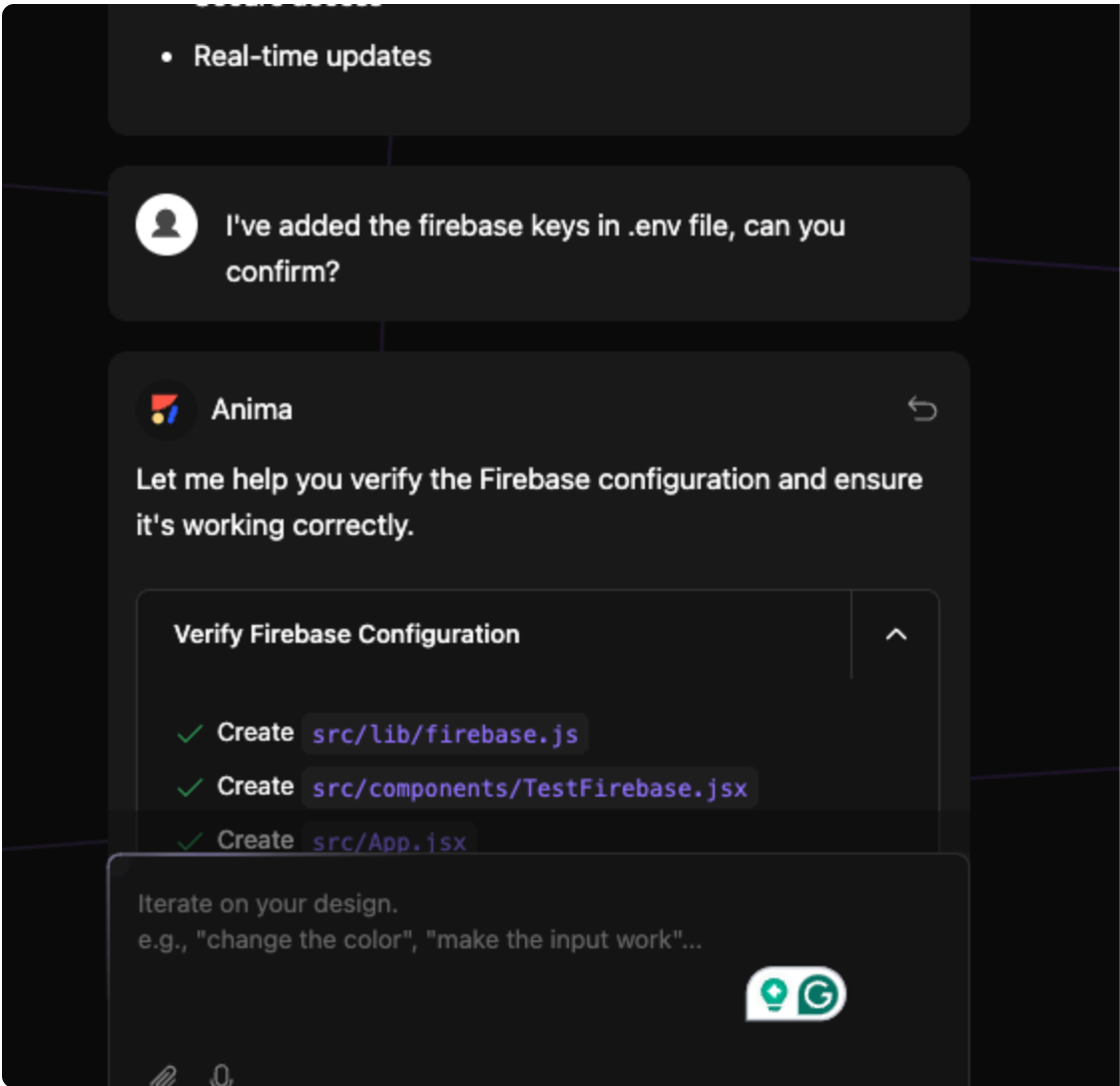
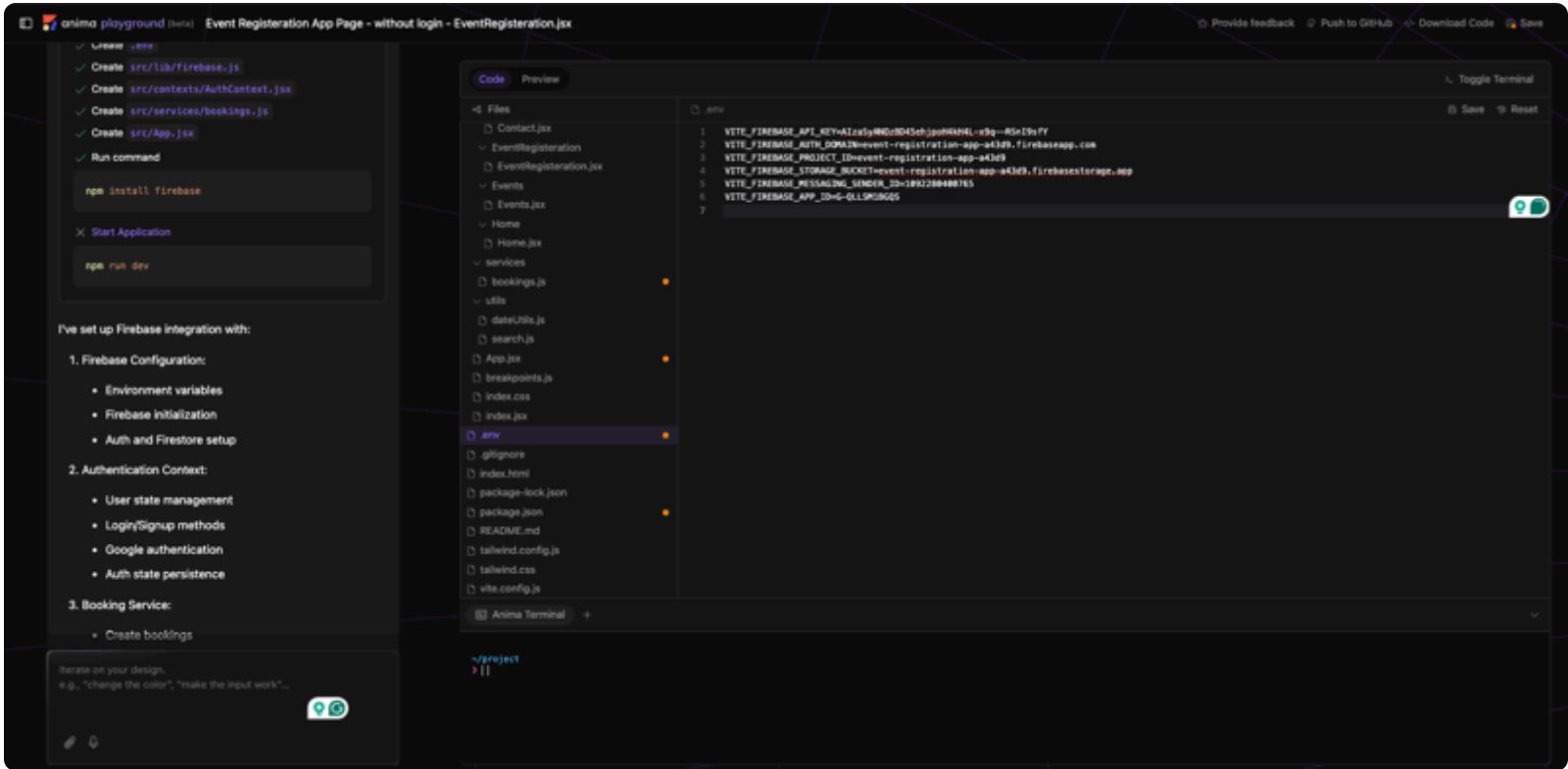
Firebase collection, and for each collection, we need a name to reidentify it again later on, similarly, we will name our collection as bookings. Each collection name should be unique in string format, meaning Booking and booking are 2 different collections in the firestore.

Now, what is firestore, simply the database which stores the data, and firestore has collections, where each collection has a unique name that stores the data and which data, any type, string,

number, boolean, uuid, object, or array.

Now we will write some code and after a long time your skills will come to use 😊, but only to edit the env file 🧑🏻💻 because we have instructed anima playground to use a .env file for importing firebase API keys.

Go to the .env file in the dev playground by clicking on the .env directory/folder in the preview tab and add the env configuration things as it is from the above firebase API keys, make sure the values remain the same, check the image below of our .env file in the anima dev playground.



Asking the dev playground about the firebase keys added in the .env file is one fun way to talk to AI to make sure everything is going smoothly.

The last step is to collect the booking data on the form submission, which we have named as the registration modal form and we will use it to prompt our dev playground.

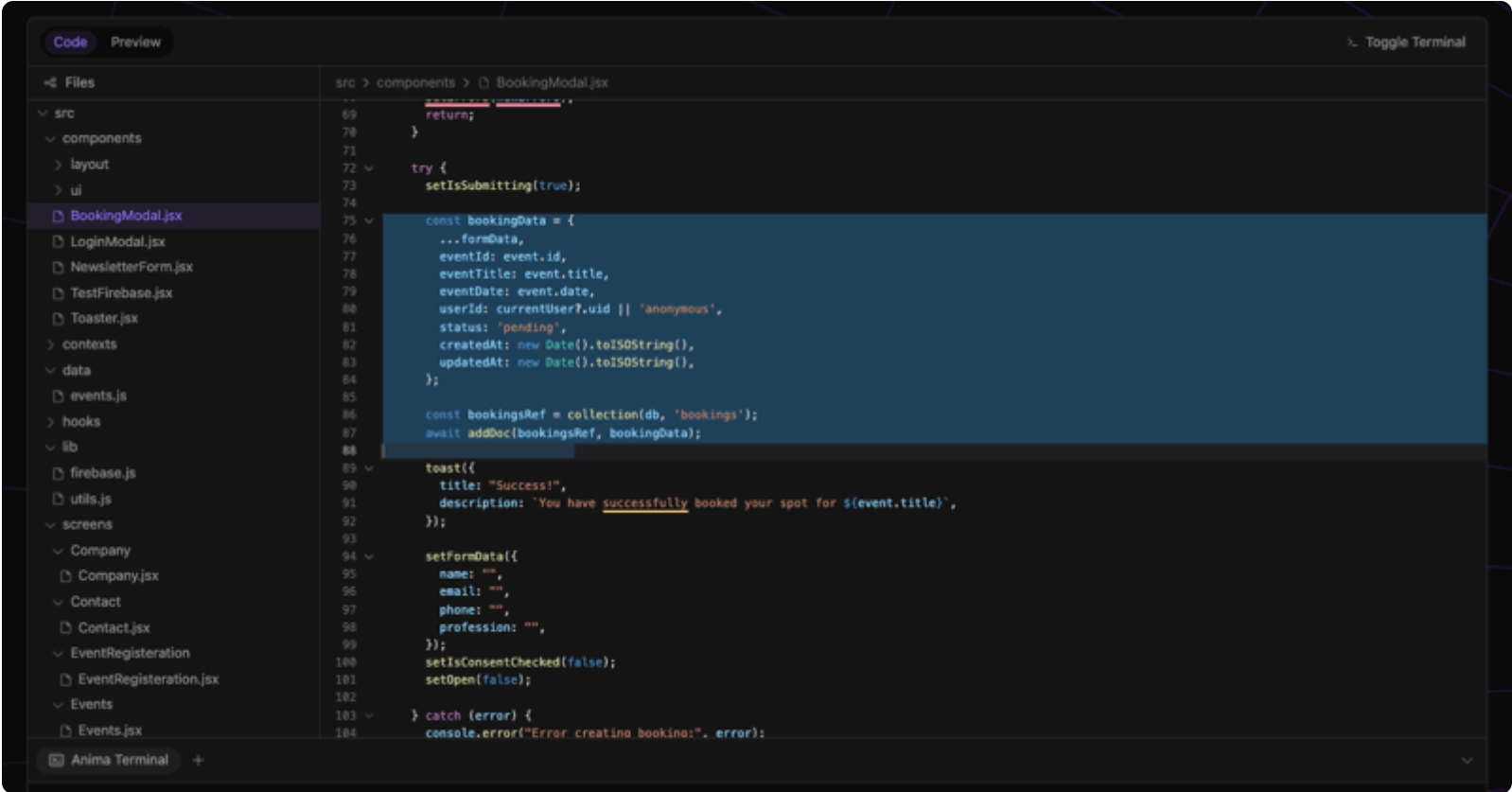
Prompt: Write a function to handle form submission from a registration modal form. On submission, collect user input data and store it in the bookings Firestore collection.

A few prompt examples,

- Generate code to capture user data from a registration modal form on submission. Save the data into the Firestore bookings collection and include error handling.



- Create a script for form submission in a React component named RegistrationModal. Use Firebase Firestore to save the collected form data into the bookings collection.
- Write code for a registration modal form. On form submission, validate the data, save it to Firestore under the bookings collection, and display a success message.



In the image shown above, the highlighted colour is the firebase method added by the anima dev playground to store the booking data along with booking details. It’s amazing how AI is writing structured and well-defined type-safe data to handle concurrency, and latency in the backend.

For the time being, we are skipping how to handle authentication and storing subscribers' data, leaving it up to you to explore and implement in your dev playground. To guide you in adding these features, here are a few helpful prompts:

- How can I set up Firebase authentication for email and password login? Provide code examples.
- Generate a function to store subscriber details like email and username in Firestore.
- Write a script to integrate Google Sign-In with Firebase authentication and store user details.
- What is the best way to validate user input before saving it in Firestore?
- How do I create a subscriber dashboard to display all saved data from Firestore?

Use these prompts to experiment, refine your implementation, and expand the functionality of your project.

## Handling Errors in Playground

Sometimes AI chatbots hallucinate or give unexpected results. Handling errors becomes important and it’s quite simple actually. If nothing works on the playground, the advice is to provide the message to the AI model

- If the AI chatbot is not taking instructions as expected, break down the instructions into chains of thoughts;
- If AI shows an error, add an error handling edge case in the prompts such as “Handle email validation and consent checkbox checked in registration modal before submitting it”

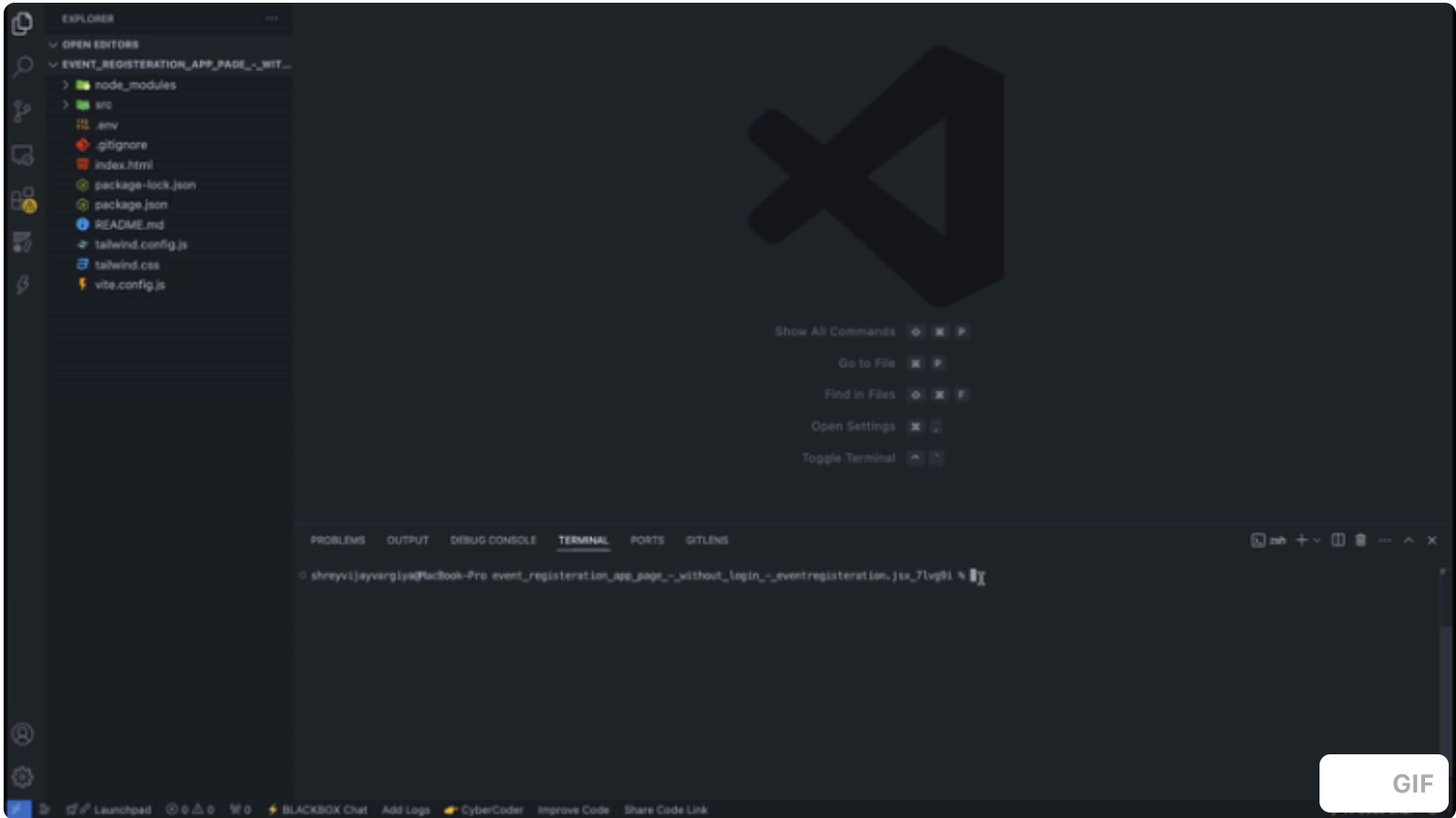
Prompting techniques will certainly help you to get better results.

## Downloading Code Repository

Once you are ready with all the final details and changes you are prepared to download the code.

Use the top right corner navbar in the dev playground and click on the “Download Code” button to download the repository.





Final code repository output from the Anima Playground app running locally

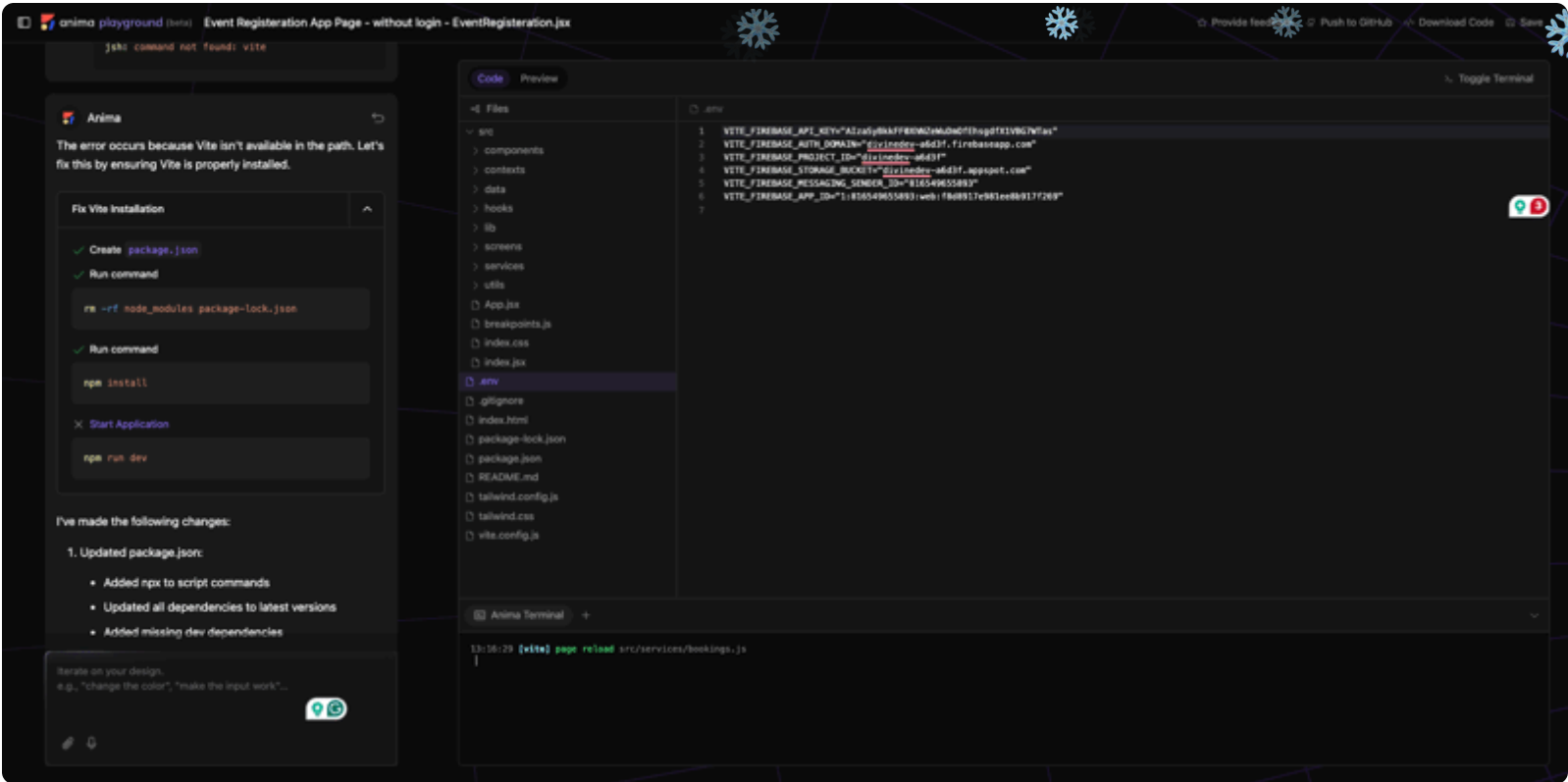
## Final results

Once that is done, below is the final list of prompts you can use to complete the event registration app page directly inside the Anima dev playground.

- Implement a login button in the header that opens and closes a modal on click. The modal should include an email/password login form and buttons for Google, Facebook, and Twitter authentication
- Implement a 'Book Now' button that opens and closes a registration modal on click. The modal should contain event details and a form for user inputs like name, email, phone number, profession, a content consent checkbox, and a submit button.
- Handle email validation and consent checkbox checked condition before submitting the registration form
- Add confirmation toast when the registration form submitted
- Complete the newsletter subscribe form feature by adding email validation and showing a confirmation toast
- Create a new “/events” page with listing company details and all events along with the “Book Now” button and registration form
- Create a “/company” page showing the company details and generate the sample company details and UI of your suggestions
- Create /contact-us page and design it according to the attached image, making it the same as the image uploaded.
- Complete the footer section by adding page URLs and social media links
- Install Firebase for data storing and authentication and import the API keys from the .env file
- Create a method to store the registration form booking in the Firebase booking collection
- Generate a method to store the newsletter subscriber data in the subscriber's collection and try with each email as the unique string
- Write a method to add authentication using email and password and other social media methods such as Google, Twitter and LinkedIn
- Write a method to send an email to [support@eventr.com](mailto:support@eventr.com) on the contact-us page and the body contains the contact-us page form elements such as email, name and message.

Feel free to adjust the timing of the prompts, and instructions, and add/delete accordingly while playing in the anima playground.

Shown below is the image of the Playground with the final code repository with all the above functionalities



You will get the repository containing the following

- Vite/React/Html code repository of your choice
- The “components” directory of shadcn UI components
- The “lib” directory contains the functions to handle tailwind merge classes
- “screens” contains the web pages such as events, account
- The root directory has all packages installed such as react, tailwind and shadcn
- All configuration files for packages such as package json, typescript, vite, tailwind configs

The entire process helps save tonnes of time, generate the exact figma design into the desired output of the developer's choice of frontend code.

## Conclusion;

Anima offers a powerful Figma plugin that seamlessly converts Figma designs into your preferred front-end tech stack. The Anima Dev Playground takes this a step further, acting as a browser-based AI coding assistant that transforms Figma designs into code directly within the Figma app, and then brings it to the Playground for further refinement. The Anima Playground boasts three core features:

1. Preview and edit code repositories.
2. AI coding assistance for enhanced productivity.
3. Download, save, or push repositories directly to GitHub.

[Try the Anima figma plugin for FREE](#)



Hostinger    PROMOTED



### [Get n8n VPS hosting 3x cheaper than a cloud solution](#)

Get fast, easy, secure n8n VPS hosting from \$4.99/mo at Hostinger. Automate any workflow using a pre-installed n8n application and no-code customization.

[Start now](#)

## Top comments (0)

[Code of Conduct](#) • [Report abuse](#)



# Learn to adapt to increasing security challenges

Join AWS Security LIVE! for expert insights and actionable tips to protect your organization and keep security teams prepared.

[Learn More](#)



## Just an artist with the keys

## LOCATION

India

## EDUCATION

Masters in Biotechnology

**JOINED**

Feb 23, 2021

More from [shrey vijayvargiya](#)

AI Chat App template — Next.js, Vercel AI SDK, Firebase

#ai #frontend #nextjs #firebase



Running deepseek locally, Ollama and langchain

#llm #openai #machinelearning #webdev



Build a Custom SaaS CRM in Hours (Not Months) with This React + Next.js Template

#saas #webdev #programming #boilerplate