



NEXTGEN CODING

INTERNSHIP REPORT

---

# Web Dev Internship

---

*Submitted By*  
Amine BENSELEM

7 juillet 2025

Table des matières

<b>1</b>	<b>React Training</b>	<b>2</b>
1.	Introduction to React . . . . .	2
2.	React Hooks . . . . .	2
3.	React Project Architecture . . . . .	2
	<b>Conclusion</b>	<b>3</b>
<b>2</b>	<b>React Project</b>	<b>4</b>
1.	Details . . . . .	4
2.	Interfaces . . . . .	4
3.	Reflection . . . . .	7

# 1 React Training

This chapter presents the key milestones and hands-on projects I completed during my learning journey with React. Through practical exercises and real-world applications, I progressively developed my skills in modern web development using React, mastering both its core principles and advanced techniques.

## 1. Introduction to React

**Objective :** Gain a solid understanding of the core concepts of React, including JSX, components, props, and state.

To get started, I studied how React works under the hood, focusing on its declarative nature and component-based architecture. I learned how to :

- Write JSX to combine HTML and JavaScript seamlessly.
- Build reusable functional components for UI sections.
- Pass data between components using props.
- Manage internal state to handle dynamic changes in the UI.

**Video resource :** Complete React Course – BroCode

**Practical work :** I created small applications such as a to-do list and a counter to reinforce my understanding of props and state.

## 2. React Hooks

**Objective :** Learn to manage component state and side effects using React hooks.

After understanding the basics, I dove deeper into React's functional approach by using hooks. I implemented :

- `useState` to update and display real-time values in the UI (e.g., counters, toggles).
- `useEffect` to perform side effects such as fetching data from external APIs and updating components based on dependencies.

**Video resource :** React Hooks – `useState` & `useEffect`

**Practical work :** I built interactive mini-projects such as a dynamic counter and an app that fetches live cryptocurrency data using public APIs. This helped me understand asynchronous operations and conditional rendering in real scenarios.

## 3. React Project Architecture

**Objective :** Learn to organize a React project for scalability, reusability, and maintainability.

As I progressed, I focused on writing cleaner and better-structured code. I applied best practices for folder structure and component separation, including :

- Creating shared and reusable components (e.g., cards, buttons, layouts).
- Structuring pages, services, and styles logically in separate folders.
- Using centralized files for API calls to make the codebase more scalable.

**Video resource :** React Project Structure – Clean Architecture

**Practical work :** I implemented a clean architecture in my own projects by organizing logic into clearly defined layers and abstracting repeated code for better readability and reuse.

## Conclusion

This training gave me a strong foundation in React development. By actively applying the concepts I learned in practical projects, I improved my ability to build responsive and interactive interfaces. I became comfortable using hooks like `useState` and `useEffect`, handling asynchronous data, integrating third-party APIs, and writing clean, modular code—all of which I applied in real-world applications such as a cryptocurrency market tracker.

## 2 React Project

### 1. Details

In this project, I developed a responsive Binance Clone using React and Tailwind CSS. The app demonstrates dynamic state management with React's `useState` ,`useEffect` hook. Tailwind CSS was used to build a clean, responsive interface that adapts well to different screen sizes. I used CoinGecko free api with a rate limit of 50 request per minute

The project source code is available on GitHub : [View the Repository](#)

Live demo is available here : [View the Demo](#)

Used technologies :

backend : `Nest Js`

frontend : `ReactJs` ,styled with `Tailwind css`

### 2. Interfaces

The website consists of four main sections, each designed to provide a clear and engaging user experience :

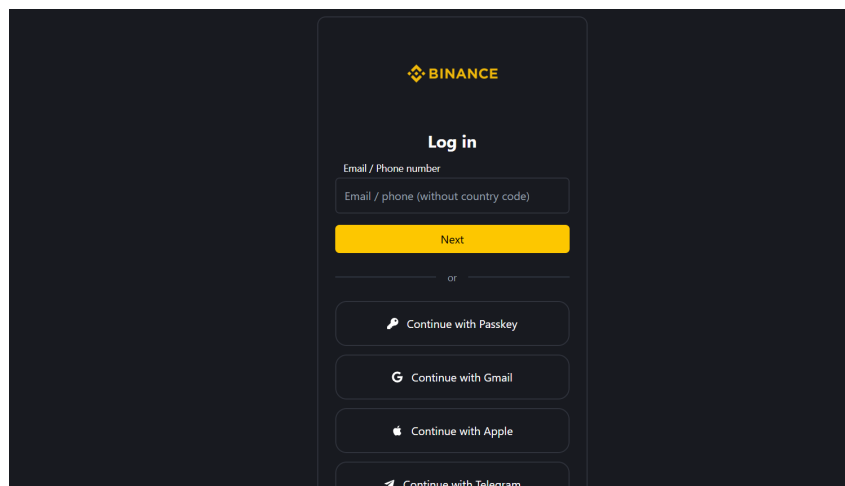
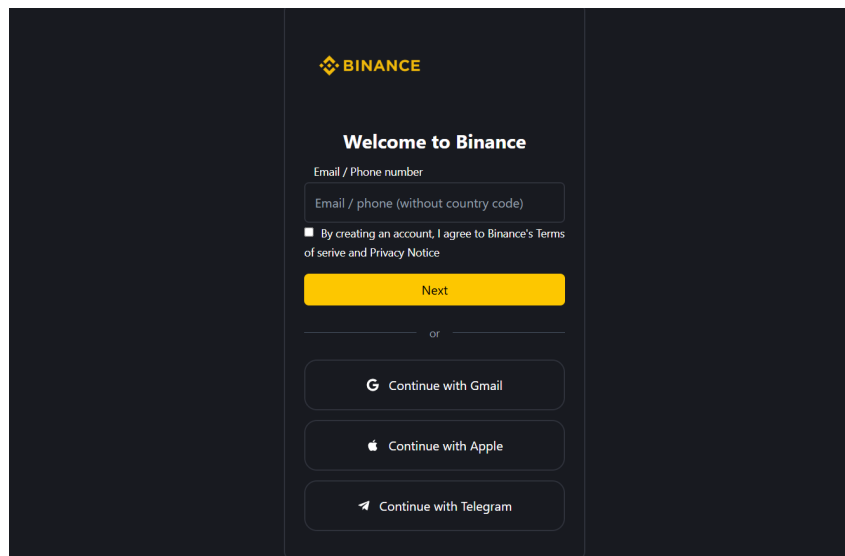
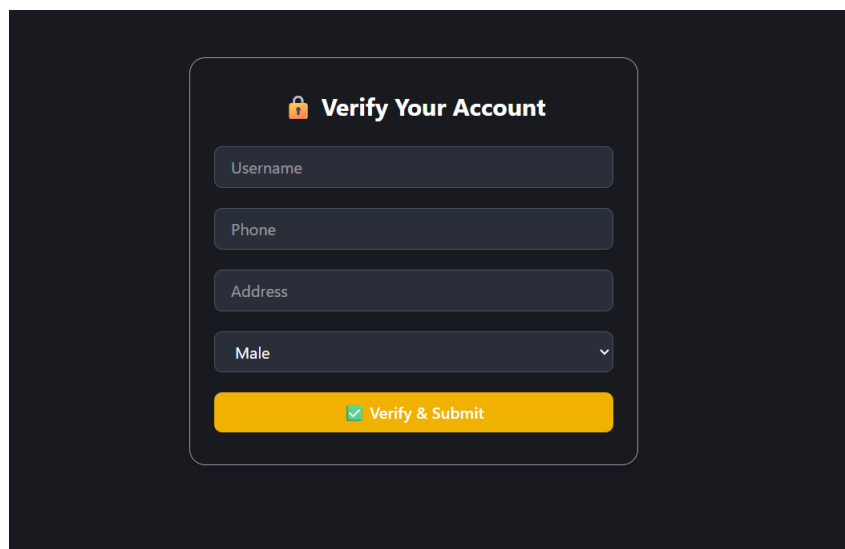


FIGURE 1 – Login page



The image shows the Binance sign-up page. At the top, the Binance logo is displayed. Below it, the heading "Welcome to Binance" is centered. Under the heading, there is a label "Email / Phone number" above a text input field. The input field contains the placeholder text "Email / phone (without country code)". Below the input field, there is a checkbox with the text "By creating an account, I agree to Binance's Terms of service and Privacy Notice". Below the checkbox is a yellow button labeled "Next". Below the "Next" button is the word "or" centered between two horizontal lines. Below the lines are three buttons: "Continue with Gmail", "Continue with Apple", and "Continue with Telegram".

FIGURE 2 – Sign up page



The image shows the Binance "Verify Your Account" page. The heading "Verify Your Account" is centered at the top, preceded by a lock icon. Below the heading are four input fields: "Username", "Phone", "Address", and a dropdown menu labeled "Male". Below these fields is a yellow button labeled "Verify & Submit" with a checkmark icon.

FIGURE 3 – Verification page

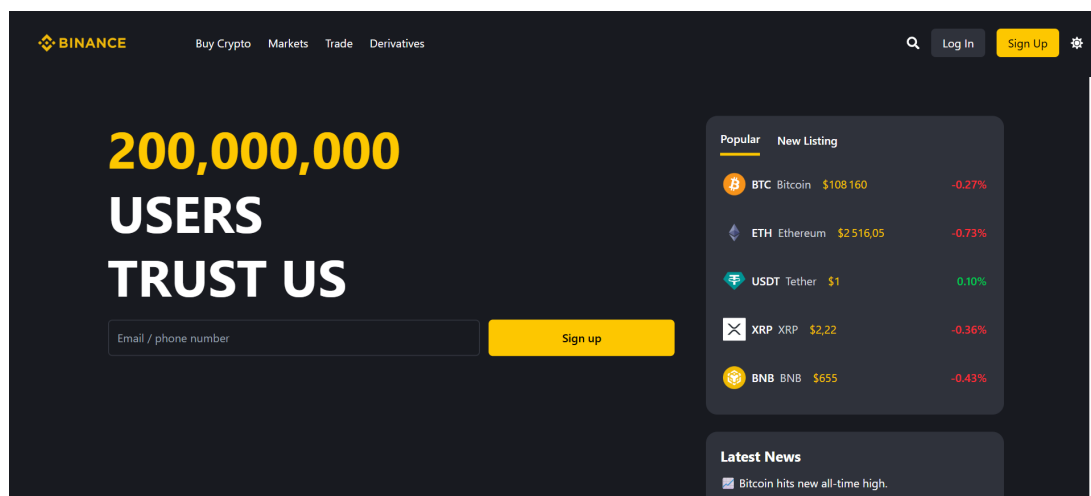


FIGURE 4 – Home Page

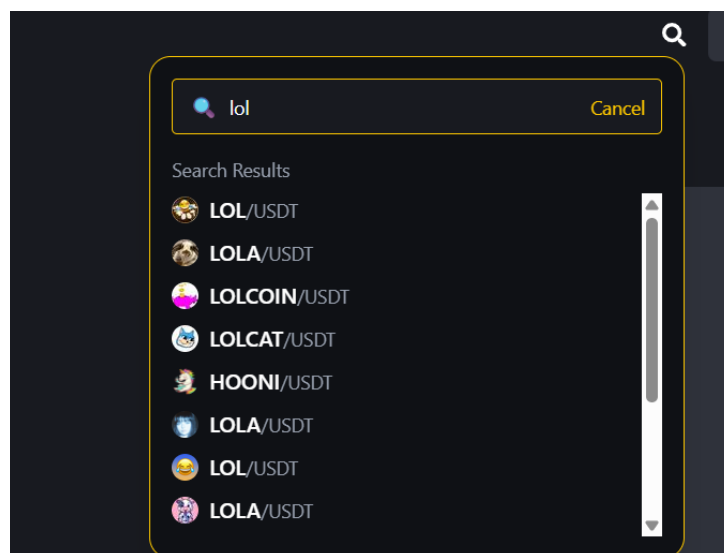


FIGURE 5 – Search Card

Name	Price	Change	24h Volume	Market Cap Rank	Actions
<b>BTC</b> Bitcoin	\$108 162	-0.08%	\$19.68B	1	
<b>ETH</b> Ethereum	\$2 515,66	-0.63%	\$12.06B	2	
<b>USDT</b> Tether	\$1	-0.01%	\$36.53B	3	
<b>XRP</b> XRP	\$2,22	-0.19%	\$1.25B	4	
<b>BNB</b> BNB	\$655,04	-0.28%	\$381.01M	5	
<b>SOL</b> Solana	\$148,04	-0.68%	\$3.11B	6	
<b>USDC</b> USDC	\$1	-0.00%	\$3.45B	7	
<b>TRX</b> TRON	\$0,284	-0.23%	\$423.3M	8	
<b>DOGE</b> Dogecoin	\$0,164	-0.71%	\$869.25M	9	
<b>STETH</b> Lido Staked Ether	\$2 514,34	-0.64%	\$8.47M	10	

Prev Page 1 Next

FIGURE 6 – Market Page

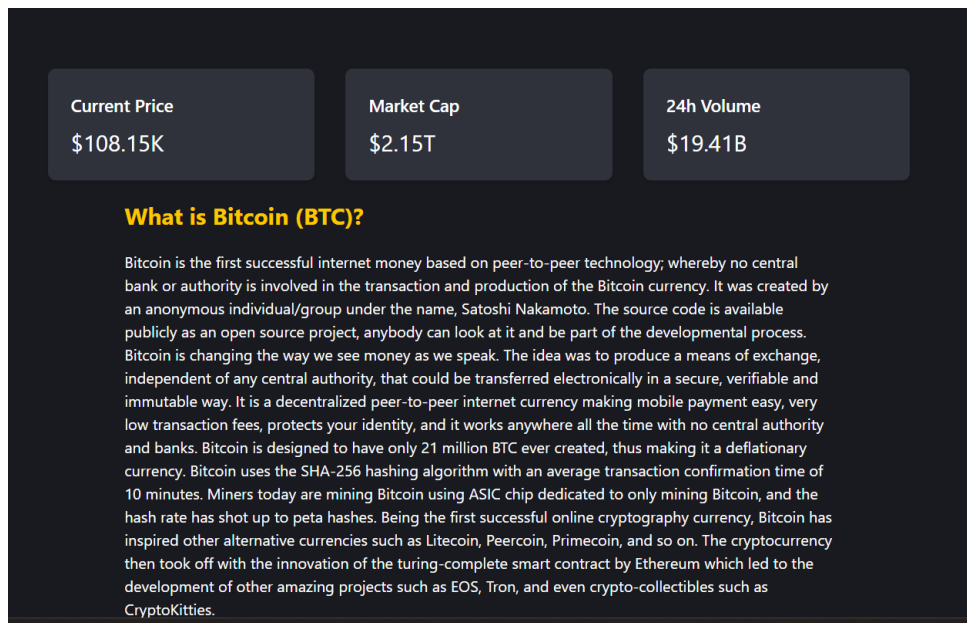


FIGURE 7 – Details Page

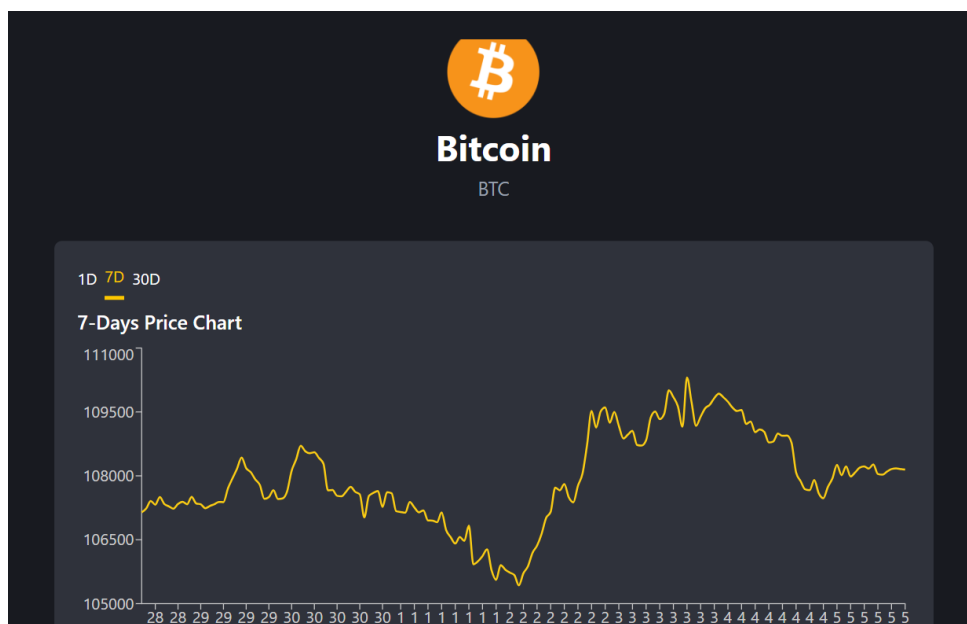


FIGURE 8 – Chart

### 3. Reflection

This project strengthened my skills in building responsive user interfaces using Tailwind CSS and in managing dynamic state and side effects in React. I gained hands-on experience with `useEffect` for consuming APIs, implementing conditional rendering based on fetched data, and handling asynchronous operations effectively.