

# Framework Choice for the Board Game Application

## Introduction:

For the development of my board game app, We added a digital experience that increase and extends the physical gameplay. The application includes the main interface and a set of minigames such as Questions, mystery boxes and side quest, all designed to work both online and offline. The app was built as a Progressive Web App (PWA) to ensure accessibility across devices and to offer users a seamless and responsive experience.

## Choice of Framework: React with Vite:

React was selected as the core framework due to its component-based architecture, which provided a structured and efficient way to manage the different screens, UI elements, and logic of the app. It allowed for reusable and modular components, making the development process more scalable and maintainable.

Vite was used as the build tool for its fast development server and optimized production builds. Compared to traditional tools like Webpack, Vite significantly reduced build times and offered a more modern development experience. Additionally, it integrates well with PWA configurations, enabling features such as offline use and app installation on mobile devices that can get added for future use.

## Use of Indexed DB (Database):

To support offline functionality, I implemented Indexed DB as the local storage solution. This browser-native database allows the application to store game data and user progress directly on the device. As a result, the app can function reliably even in low or no internet connectivity scenarios.

Indexed DB was particularly suitable for this project because:

- It enables persistent storage of gameplay data across sessions.
- It improves user experience by maintaining functionality regardless of network status.
- It supports larger datasets and structured storage, ideal for tracking game elements like, unlocked content, and progress.

## Suitability of the Stack:

This technical stack—React, Vite, and IndexedDB—was well-suited to the needs of this project for the following reasons:

- **Efficiency:** Fast development and build times with Vite.
- **Scalability:** React's component structure supported complex UI requirements.
- **Offline Capability:** IndexedDB allowed for uninterrupted use of the app.
- **Cross-platform Accessibility:** The PWA setup ensured that the app could run on various devices, including tablets, phones, and desktops.

This tech stack enabled the development of an application that was not only functional but also scalable and adaptable for future enhancements.

## Reflection:

This project helped me strengthen my skills in building interactive web applications using modern tools. React and Vite made development fast and efficient, while Indexed-DB gave me insight into managing offline data. I learned how to structure an app for both usability and performance, and how to solve technical challenges, especially around data persistence. Overall, the experience was valuable for both my technical growth.

## Resources:

React Team. (n.d.). *React: A JavaScript library for building user interfaces*. Meta Platforms, Inc. Retrieved June 16, 2025, from <https://reactjs.org/>

You, E. (n.d.). *Vite: Next generation frontend tooling*. Vite. Retrieved June 16, 2025, from <https://vitejs.dev/>

Contributors. (n.d.). *IndexedDB API*. MDN Web Docs. Retrieved June 16, 2025, from [https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB\\_API](https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API)