

# Breno Brasil

FULL-STACK SOFTWARE DEVELOPER

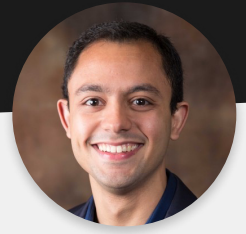
📞 (615) 573-8950

✉️ [contact@brenob.com](mailto:contact@brenob.com)

🌐 [www.brenob.com](http://www.brenob.com)

🐙 [github.com/brenobrdev](https://github.com/brenobrdev)

🌐 [linkedin.com/in/breno-brasil](https://linkedin.com/in/breno-brasil)



## My Developer Pathway

I come from a multi-cultural background, growing up in a small town in Brazil, and moving to the U.S. for college. I graduated with a degree in Molecular Biology, and currently work as a Cancer Research Data Coordinator. I always had a fascination with computers, but since my focus throughout college was to go to medical school, I did not explore coding. After I finished college, I finally started coding, just as a fun thing to do in my free time, but it quickly turned into an activity I enjoy and want to build a career with. I thoroughly enjoy the process of building software from scratch and engaging with complex problems that I can solve and learn from. I continue to use my free time to learn more about coding and develop my skills as a programmer. I look forward to the opportunity to further develop my skills in a professional setting while contributing to the empowerment of the team I work with.

## Skills

HTML | CSS | Javascript | Git | GitHub | SQL | C# | VSCode

## Personal Projects

**Organization Tool** - HTML | CSS | Javascript

Application that I built to assist me in my current job, to help me keep track of the many different tasks and notes, and help me do my work more efficiently. It has two main parts, a list of tasks, and several blocks for notes.

- Created a data structure for tasks, in that each task is an object that contains a state (done or pending), a clinical trial (study), a category, a patient, a description, and a due date.
- Ability to sort tasks manually (drag and drop), or sort by study, category, or due date.
- Implementation of CRUD operations where all the data (tasks and notes) is saved on the browser's local storage automatically, as the program is used. Each data point in a task can be modified or deleted.
- All data is stringified in order to be saved into local storage, and parsed into JSON to be able to be used in the application.
- Local storage was preferred due to administrative rights on the system.
- Used CSS Grid and Flexbox to build the design.

**Guess The Word!** - HTML | CSS | Javascript

Hangman-style game coded in two versions: English and Brazilian Portuguese. The purpose of this project was to practice my skills and for my brother to play it.

- A challenge was to organize the code to accommodate two different languages, and at the same time allow future implementations of other languages. On the code side, all translation happens within a single method.
- The game will randomly select a word from the dictionary of the selected language, and display one "\_" for each letter in that word. The player has seven tries to guess the correct word by selecting individual letters. If a player selects a letter that is not part of the word selected by the game, a part of an SVG character is displayed.
- State management was implemented so that once a letter is used, it is inactivated and the player cannot select it again.

**Pig Game** - HTML | CSS | Javascript

Game idea proposed in one course I took. I did not follow the tutorial, and I did not use any of the starter code provided. I wanted a challenge, so I decided to build the interface and the logic of the game from start to finish by myself.

- A challenge was to use the principle of separation of concerns to organize the code, trying to keep each method responsible for only one functionality of the game logic.
- State management was used to determine the player's turn.
- Two-player game where each player has a "Current" score and a "Total" score.
- A player rolls the dice, and if the result is different than 1, the number on the dice is added to the player's "Current" score. The player continues to roll the dice until the dice result is 1, or until the player decides to hold.
  - If the dice result is 1, the current player loses all the points in the "Current" score, and passes the turn.
  - If the player decides to hold, all the "Current" score is added to the "Total" score, and the player passes the turn.
- The first player to reach a "Total" score of 100, wins.