

CipherGPT

Brogrammers
Github Team: CipherGPT

Roan Morgan, Jonathan Attanasio

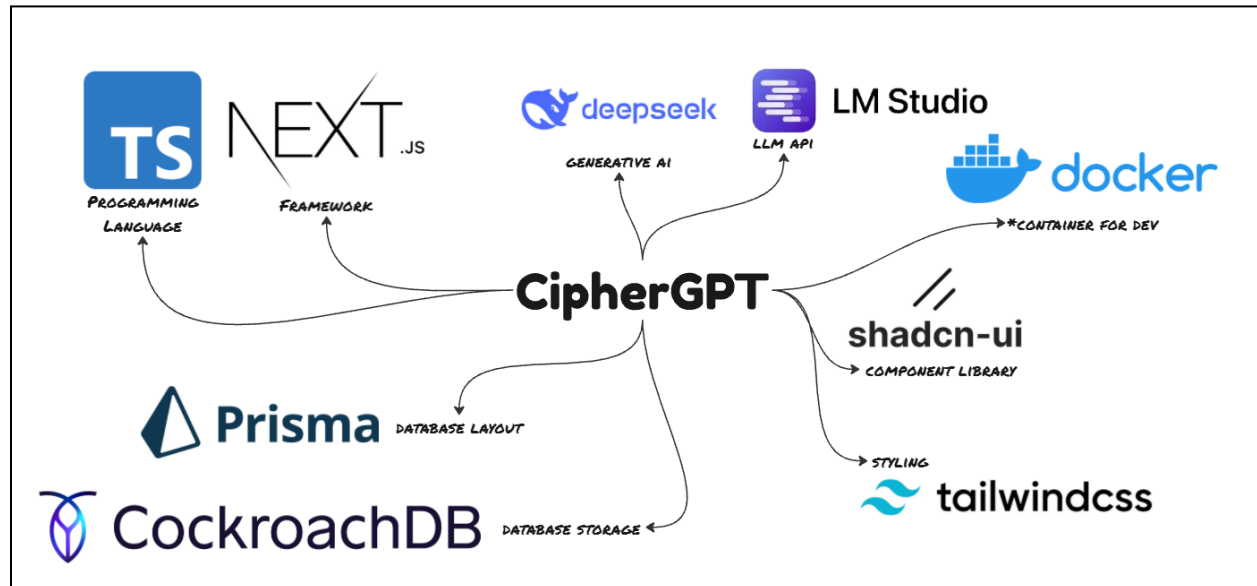
Introduction:

CipherGPT is a web application that leverages full stack development to make an in-browser interactive game. This web app will include [telephone](#) style mini-games where a Large Language Model (LLM) plays as the middle man construing messages. These messages are then conveyed to the player that is given a task to complete with the information conveyed by the other player. This app idea comes from our desire to strengthen our skills with full stack development and how local AI models can be integrated and leveraged to make a consumable product. Our game presents novelty in the form of using LLMs for entertainment. It is no secret that LLMs have already been used to supply generated text dialogue for characters in some new video games. These games are however slightly niche and do not provide an inviting user experience. This game will be accessible to anyone who can browse the internet, making it consumable to a larger audience. Jon has an extensive background in front end development and a passion for making a streamlined user experience. Roan has an affinity for learning about the backend and an interest in integrating local AI models into consumable products.

Customer Value:

The primary customer is going to be anybody that is familiar with LLMs or anybody curious as to how they can be used for more fun things than cheating in academia. The customer wants an easy to use interface with quick response times no matter the application. This does not inherently solve any problem for our users. Our solution will deliver a fun way to interact with generative AI in a game-like environment with leaderboards for each game. Everyone that we've voiced the idea to seems to love it. Similar to the New York Times games they are fairly simple, the difference is that all of our's are based around generative AI. Our customer-centric measures of success would be that our leaderboard fills up because people keep coming back to play over and over, that would quantify success in a measurable way.

Proposed Solution & Technology:



- **Framework (Next.js):** A React-based framework that enables server-side rendering, static site generation, and API routes for scalable web applications.
- **Programming Language (TypeScript):** A statically typed superset of JavaScript that enhances code maintainability and developer productivity.
- **Database Layout (Prisma ORM):** A modern database toolkit that provides type-safe database queries and schema management.
- **Database Storage (CockroachDB):** A distributed SQL database designed for high availability, scalability, and strong consistency.
- **Generative AI (Deepseek):** An advanced AI model for text generation, coding assistance, and content creation.
- **LLM API (LM Studio):** A local inference platform for running and fine-tuning large language models efficiently.
- **Component Library (ShadCN UI):** A collection of customizable and accessible UI components built on Radix and Tailwind CSS.
- **Styling (Tailwind CSS):** A utility-first CSS framework for rapidly building modern and responsive user interfaces.

***Container for Dev (Docker):** Docker is a container for application development and at this time we are still deliberating whether or not it is necessary to use.

Team:

Roan will be working on the backend of the stack, working to integrate an LLM into the app and fine tuning the LLM to be a welcome challenge but not too difficult as to be impossible to win the game. Jonathan will be responsible for the front side of things. Working to make an aesthetic and easy to use interface for the user. The hardest thing frontend wise will probably be tying in

the LLM. Ensuring that everything is tied together from user to backend will be one of our team's largest collaborations.

Project Management:

With the tools available, it is possible and feasible to get a working model of our app working within the given time frame. We will meet weekly for meetings most often through discord

Tentative Week-by-Week Schedule

Week of 2/16/25

- Become comfortable and versed in the technologies used

Week of 2/23/25

- Begin developing first app iteration

Week of 3/02/25

- Continue developing first iteration

Week of 3/09/25

- Complete first iteration

Week of 3/16/25

- SPRING BREAK

Week of 3/23/25

- Bug fixing and perfecting of first iteration

Week of 3/30/25

- Begin adding new features that make the user experience more dynamic and enjoyable

Week of 4/13/25

- Finish second iteration

Week of 4/20/25

- Debug and perfect

Week of 4/27/25

- Launch and prepare for assessment

Week of 5/04/25

- Celebrate

Constraints

As of now, no constraints have presented themselves when it comes to this project. If there are any, they will present themselves during the development process.

Resources

Not much is needed in the form data for this project, as we will not be requiring data.

Descoping

If the main goal for the project does not come to fruition, we would have to pivot our functionality and gimmick. Without the LLM working as the middle man, the project would be completely out of operation. This product would not be a useful entertainment medium in that case.