OiStir

Stirring up pearls of inspiration

Website Name:

OiStir

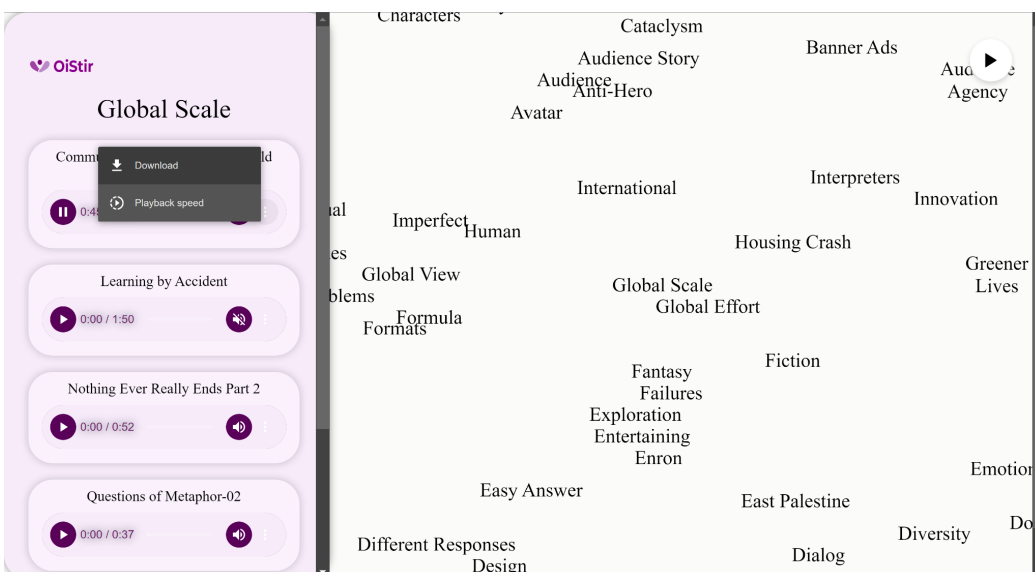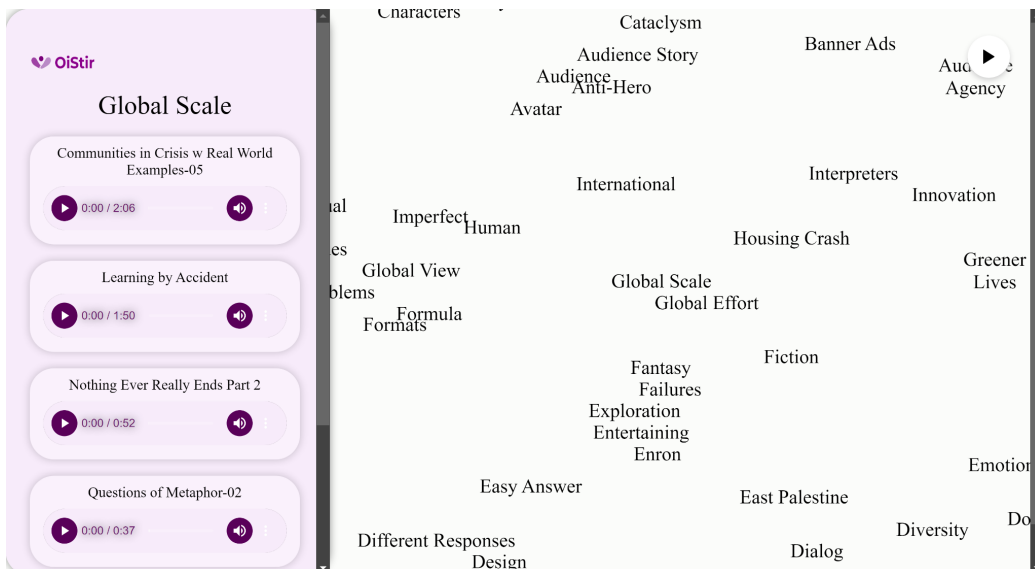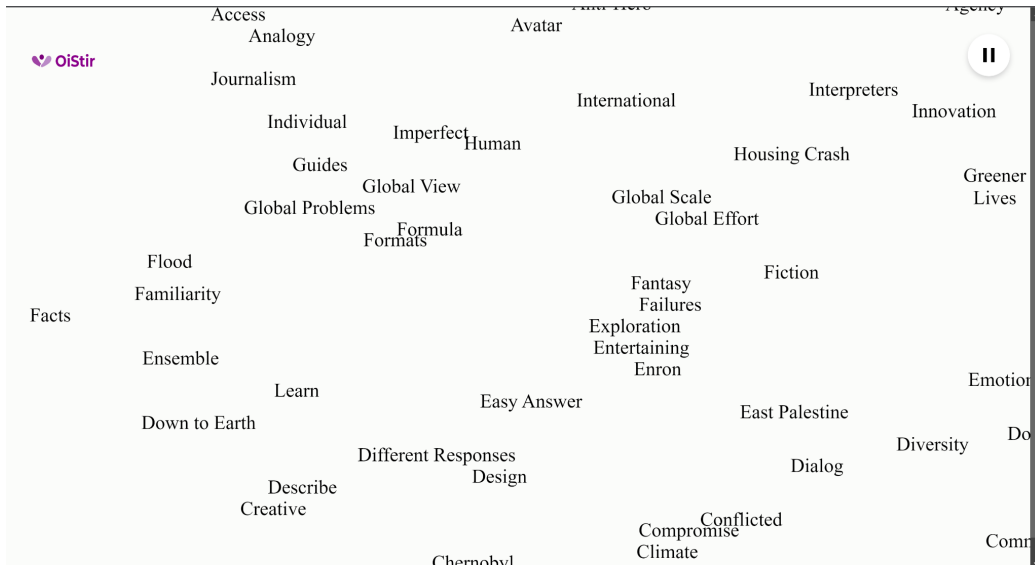# Motto:

Stirring up pearls of inspiration

Logo:

Logo + Wordmark:

Lock-up:

Screenshots:

Project Description:

A brainstorming aid used by professional film/TV writers, content developers and screenwriters who want to jumpstart their creative process to produce films surrounding today's climate issues.

Target Audience:

Those wanting to tell a story about the climate crisis but don't know where to start or want to come up with a concept that has not yet been explored.

Design:

The website's unique design was created with purpose. In order to break the limitations of standard website design and invent something outside of the mold. The dynamic screen and avoidance of hierarchy in the words appearing on the screen helps the user feel as though they are experiencing change encounters with these ideas, refraining from "giving an answer" the way lists often do.

Key Terms

- Brainstorm
- Ideation
- Audio
- AI
- Screenwriting
- Screenwriter
- Entertainment
- Fine arts

Back End Description:

Github includes commented code and a ReadMe file to allow for easy navigation and running of the website application.

```
const RainWords = ({ words }: RainProps) => {
  // Array of objects that represent the state of each falling word
  const [wordStates, setWordStates] = useState<any[]>([]);
  //Currently selected word that the user clicked on
  const [selectedWord, setSelectedWord] = useState("");
  //Boolean flag that indicates whether the sideBar is currently open or not
  const [isSidebarOpen, setIsSidebarOpen] = useState(false);
  //Window height + width are height and width of the browser window
  const [windowHeight, setWindowHeight] = useState<number>(0);
  const [windowWidth, setWindowWidth] = useState<number>(0);
  const [isAnimationPlaying, setIsAnimationPlaying] = useState(true);

  //Called whenever the array of words changes, and it it initializes a wordStates array with some values
  useEffect(() => {
    setWordStates(
      words.map((word: any, index: number) => ({
        top: -50,
        left: Math.random() * window.innerWidth,
        delay: index * 0.3,
        shift: Math.random() * 100 - 50,
      }))
    );
  }, [words]);
```

```
Getting Started

First, run the development server:

    npm run dev
    # or
    yarn dev
    # or
    pnpm dev

Open http://localhost:3000 with your browser to see the result.

You can start editing the page by modifying pages/index.tsx. The page auto-updates as you edit the file.

API routes can be accessed on http://localhost:3000/api/hello. This endpoint can be edited in pages/api/hello.ts.

The pages/api directory is mapped to /api/*. Files in this directory are treated as API routes instead of React pages.

This project uses next/font to automatically optimize and load Inter, a custom Google Font.
```

The audio clips and their respective names are stored within firebase in the firestore database and storage utility.

Each audio clip is saved within a collection name that instantiates the main topic of that particular discussion. Words can have multiple audio files associated with them.