Week 1. Introduction to Creative Web Dev.

# **What is Cursor AI?**

* 1. An AI-Integrated Code Editor
  2. Before Cursor AI, VS Code (Visual Studio Code) was commonly used
  3. After the introduction of LLMs (ChatGPT, Claude AI), many developers started to integrate LLMs within their programming
  4. Cursor AI significantly improves Developer UX within this integration
  5. Relatively new, the next ‘Game Changer’ might change within a couple of months
  6. Cursor AI = Your Editor (Where you code)

# What is Next.js?

* 1. React.js: State-of-the-Art Frontend Framework developed by Meta
  2. Limitation of React: Frontend-only tool, relatively low-level/too simple
  3. Introduction of Next.js: React-based + Server-side-Rendering + SEO + Other optimisations….
  4. Next.js: App Router vs. Pages Router
     1. App Router: Next.js 13+, Server-based logic
     2. Pages Router: The old way of doing Next.js, Client-first logic
     3. We’ll stick on Pages Router due to…
        1. Easier learning curve
        2. Easier integration with Socket.io
     4. Disclaimer: When looking for Tutorials & Googling for resources, please do consider if the code is written in App Router or Pages Router

# Git clone the repository

* 1. <https://github.com/ericggul/exp-exp>
  2. Open up your parent file directory on Cursor AI
  3. git clone <https://github.com/ericggul/exp-exp.git>
  4. Open up your exp-exp file directory on Cursor AI
     1. Exp-exp: Experimental Experiences
     2. This is the place where we’ll host example codes/in-class coding sessions within next three weeks

1. **Download the dependencies**
   1. What is a Package Manager?
      1. Packages: Code that someone else had written for you
      2. You can download different ‘packages’ via Package Manager
      3. Examples: Three.js, Tone.js, Socket.io, ….
   2. npm vs. yarn vs. pnpm
      1. npm: Default & Heavy
      2. **yarn: Lighter & Secured**
         1. Let’s use yarn for stability purposes
      3. pnpm: New Fast package manager saving disk space
   3. Terminal Line Commands
      1. yarn
         1. Installs all packages written in the package.json
      2. yarn add <package name>
         1. Installs designated package
      3. yarn dev
         1. Kicks off your localhost, now ready to code!
2. **Hello World~!**
   1. Navigate localhost:3000 on your browser
      1. Chrome as default
      2. Say hi to the ‘Hello World’!
   2. Where is this code coming from?
      1. pages > index.js
      2. localhost:3000
   3. How do you navigate to other URLs?
      1. Pages > id412 > w1 > time-interval > 1.js
      2. localhost:3000/id412/w1/time-interval/1
      3. Components > id412 > w1 > time-interval > 1 > index.js
      4. Note: We will mainly focus on id412 file directories
         1. pages > id412
         2. components > id412
   4. What are each directory for?
      1. components > Where most of things happen, where you code UI
      2. node\_modules > Packages codes downloaded from yarn
      3. pages > Where you handle Routes/Pages
         1. pages > api > Server side api calling
      4. public > Where you store your assets (images, videos, 3d models)
      5. styles > Style settings
      6. utils > Utility functions (React’s Custom Hooks)
   5. Components code structure
      1. Each folder has index.js and styles.js
         1. index.js: Your logic and JSX components
         2. styles.js: Styled-components powered styling

Now let’s start coding!

* Time interval

1. Simple Time Interval (ex. localhost:3000/id412/w1/time-interval/1)
2. Miyajima Tatsuo-inspired Time Interval [LINK](https://www.mca.com.au/exhibitions/tatsuo-miyajima-connecting-with-everything/)

* Mouse move

1. Simple Mouse Move Visual (ex. localhost:3000/id412/w1/mouse-move/1)
2. Simple Kaonashi
3. Scary Kaonashi
4. James Turrell-inspired interaction [LINK](https://www.pacegallery.com/artists/james-turrell/)

* Image

1. Simple Image
2. Multiple Images
3. Grid Images
4. Overlap Images
5. Chaotic Images

* Video

1. Simple Youtube iframe (Kim Beom [LINK](https://artreview.com/kim-beom-do-not-think-at-all/))
2. Youtube iframe with designated start time
3. Multiple Youtubes
4. Overlap Youtubes

* Data

1. Simple Data Calling
2. Data with Date
3. Simple Data Vis

* Word-puzzle (Beta)

1. Simple Word Puzzle
2. Experimental Typography
3. **Deployment**
   1. Deploying: Making your localhost-coded web-app live, accessible throughout the world!
   2. Next.js → Easy Deployment via Vercel
   3. Sing-up Vercel <https://vercel.com/>
   4. First, you need to upload your code to the git
   5. Github: Create a new repository <https://github.com/>
   6. On your Cursor AI Terminal (You might have to add a new terminal), type
      1. git add .
      2. git commit -m “Update”
      3. git remote remove origin
      4. git remote add origin **YOUR GITHUB REMOTE ORIGIN**
      5. git push -u origin main
   7. Now you have your code repository on your github
   8. Head to vercel <https://vercel.com/>
   9. Add New > Project > Import Git Repository > ‘Import’ > **Framework Preset set Next.js >** Deploy
   10. Tada!
   11. Disclaimer: You CANNOT use Vercel Deployment if you use socket.io (which we will learn next week). In this case, you’ll have to use Heroku Deployment or other alternatives. Heroku Deployment is usually not free, but you can register for the student discount. The deployment workflow will be pretty much similar, lots of resources in google.