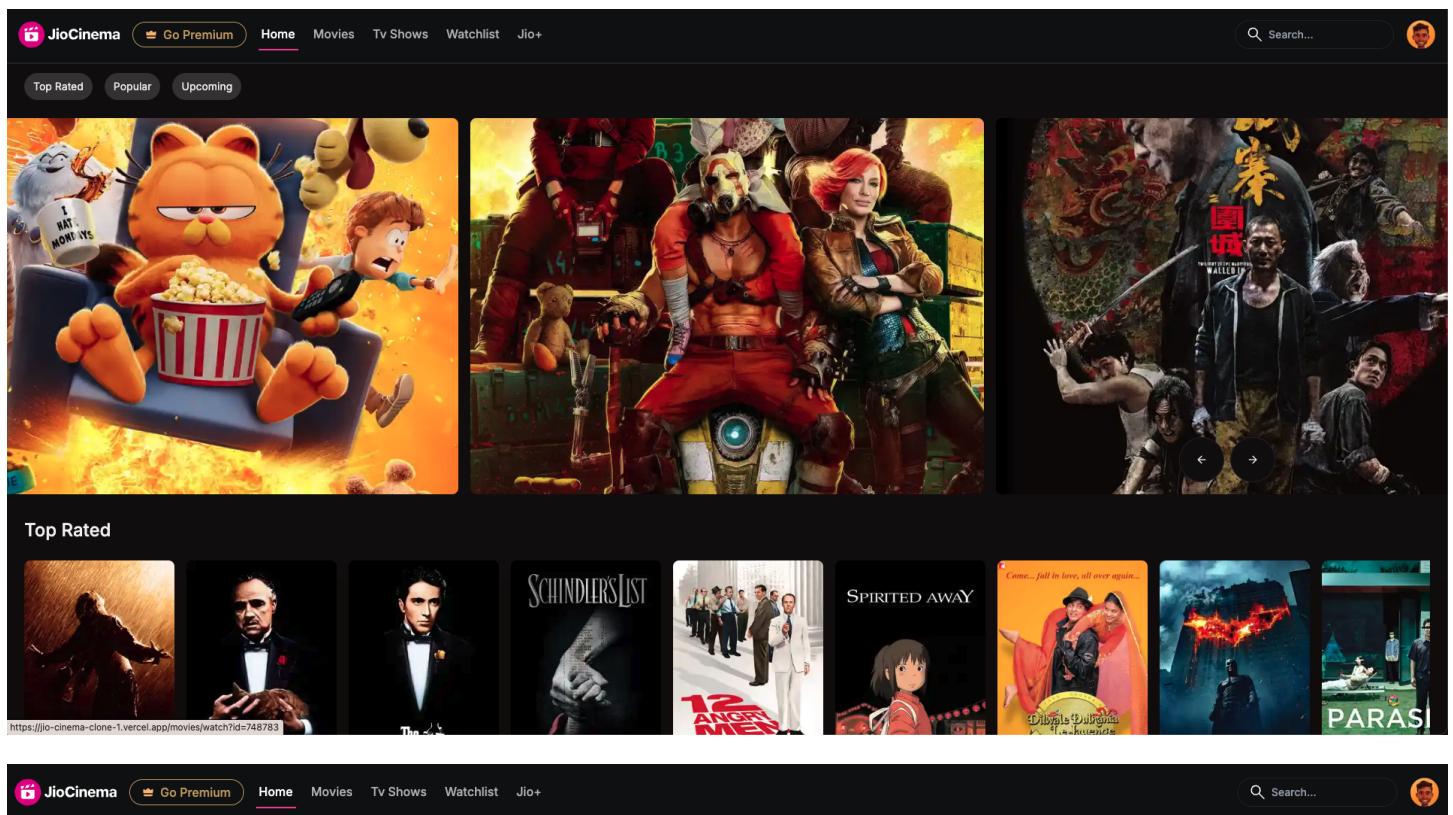# 1_JioCinema (Intro to NextJs)

Agenda

- WalkThrough of Project
- What is meta framwork -> Nextjs
- React vs nextjs
- setup of nextjs
- core concepts of nextjs

WalkThrough of Project

The UI is as same as the real JioCinema app.

In the top navbar we have options:

- Home
  - Top Rated
  - Popular
  - Upcoming
- Go Premium

- Movies
  - Top Comedy Movies
  - Top Horror Movies
  - Top Romance Movies
  - Top Action Movies
- TV Shows
  - Comedy
  - Crime
  - Drama
  - Action
- Watchlist
- Jio+
- Search
- Guest Profile
  - Login
  - Sign Up

Tech Stack

 - Nextjs

- TailwindCSS
- Component library -> ShadcnUI
- Redux -> upgrade to context API
- payment -> Razorpay
- streaming: Nodejs
- forget, reset password: mail send
- TMDB API: Movies, TV shows
- Deployment -> vercel (frontend), `render` -> (API host)

ShadcnUI : `Component library` A collection of pre-built UI components designed for Next.js and Tailwind CSS.
Redux ToolKit: A state management library for JavaScript applications. Helps manage global application state in short upgraded vserion of context API
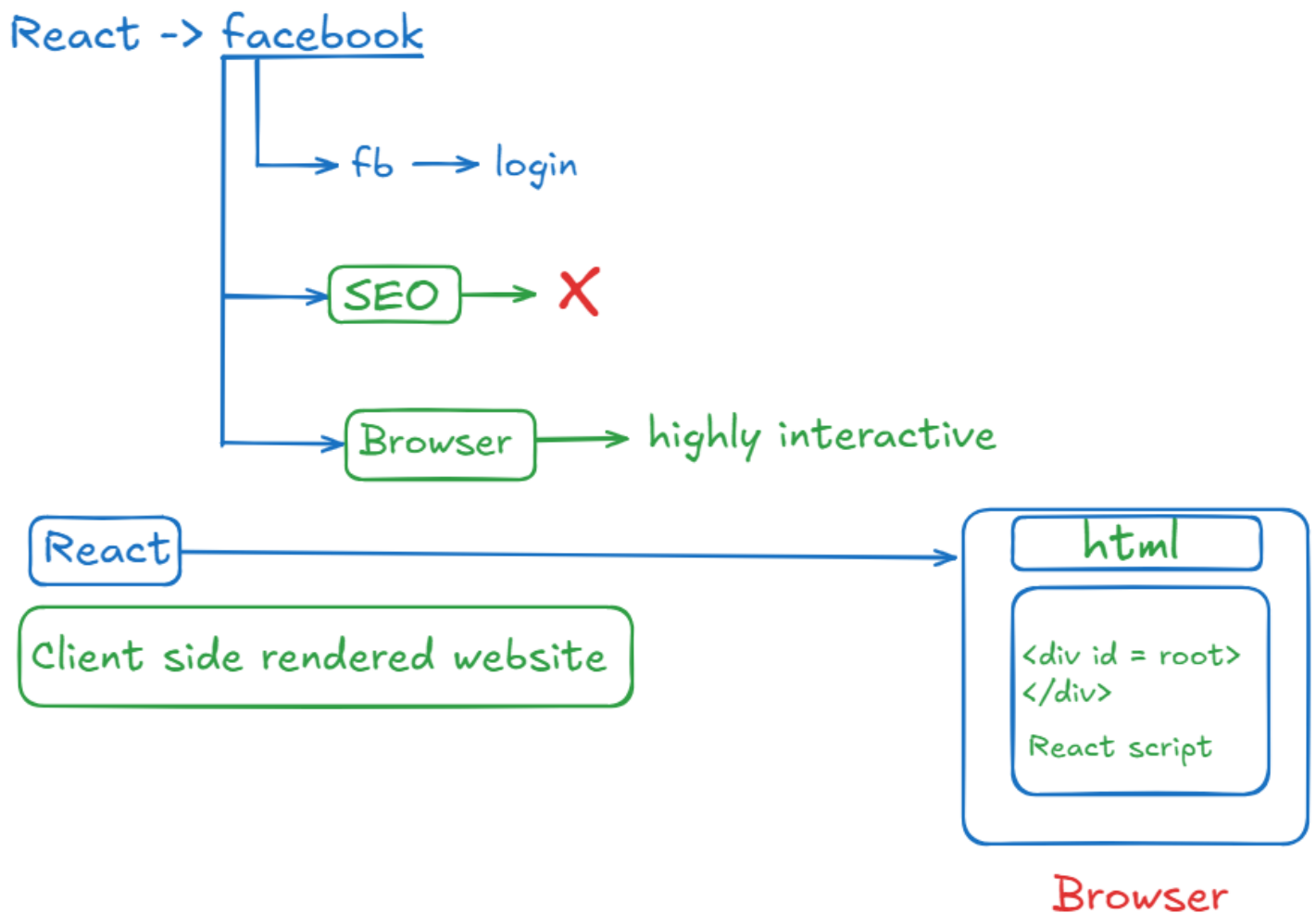**Razorpay**: payment gateway that enables secure online payments that supports various

payment methods, including credit cards, UPI, and net banking.

Streaming (Node.js): Node.js is often used for building streaming servers due to its efficiency and scalability.

Forget/Reset Password (Email): A feature that allows users to recover their passwords if they forget them via email

How react sends

React -> facebook

fb → login

SEO → ✗

Browser → highly interactive

React → html

Client side rendered website
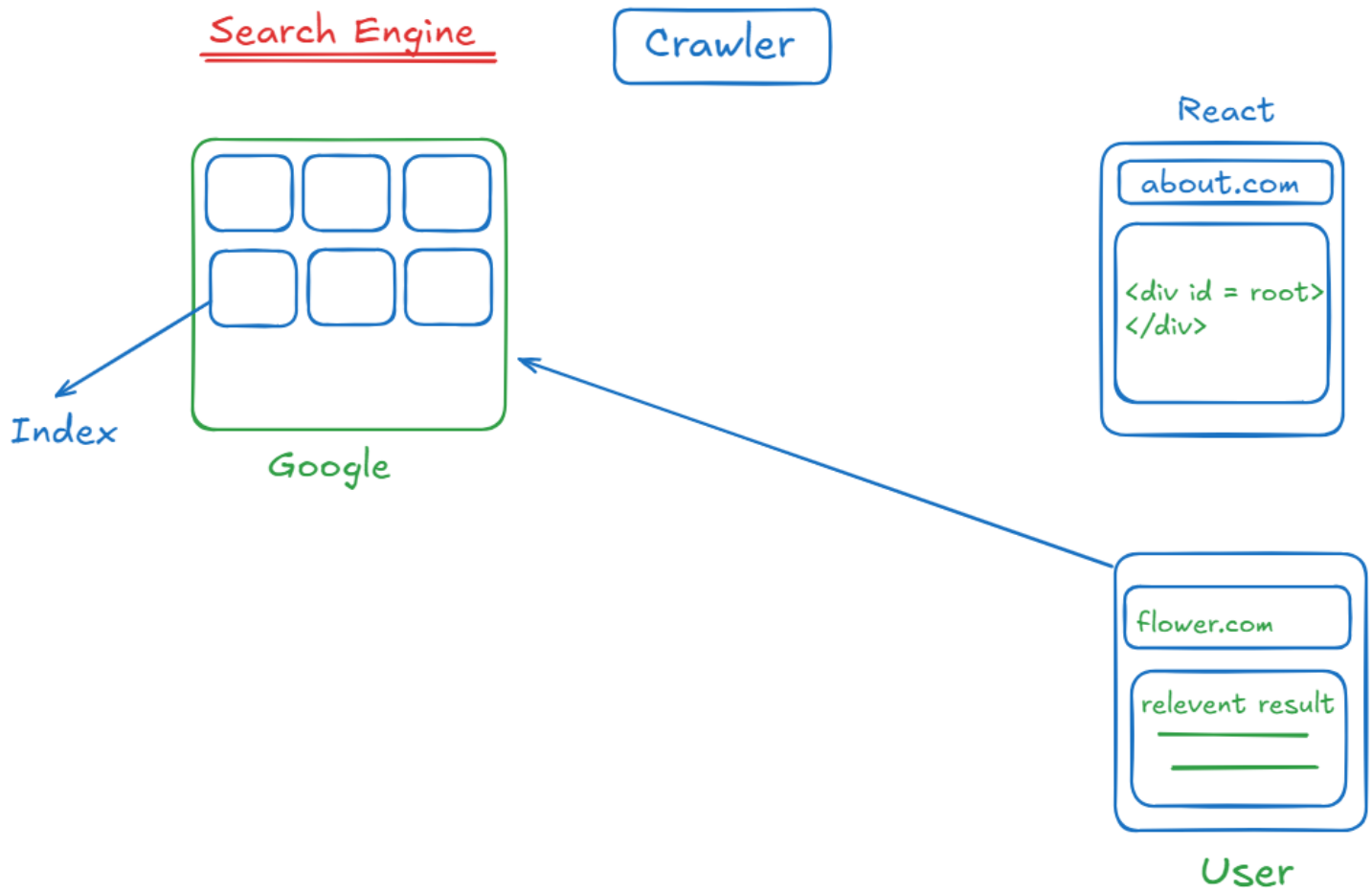
```
<div id = root>
</div>

React script
```

Browser

- It only sends empty html page to the frontend
- This type of application is known as client side rendered website. they are bad at SEO

SEO

SEO (Search Engine Optimisation) is the practice of improving a website's visibility and ranking in search engine results pages (SERPs) like Google, Bing, and Yahoo. The goal is to increase organic traffic to your website by making it more relevant and user-friendly for search engines.

Indexing is the process by which search engines store information about web pages in their databases. When a search engine's crawler discovers a new web page, it analyzes the content and adds it to its index. This allows the search engine to retrieve and display relevant pages when users perform searches.

Crawlers (also known as spiders or bots) are software programs that search engines use to discover and index web pages. They follow links from one page to another, exploring the internet and building a vast database of websites.
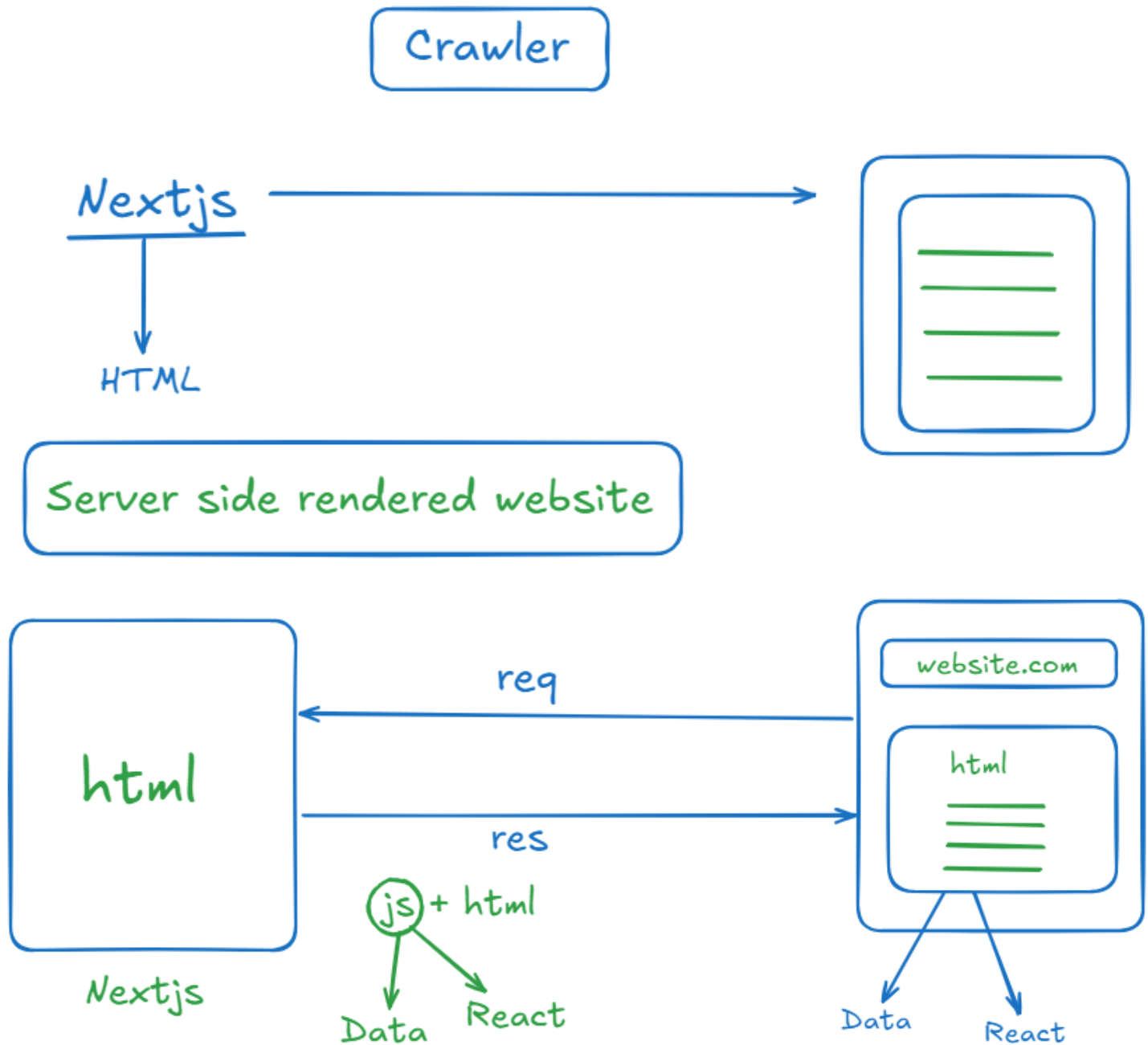

NextJs

Next.js is a popular React framework that simplifies the development of server-rendered React applications. It offers several key features that make it a preferred choice for many developers.

Initial load time reduces

- SEO improve
- Inbuilt routing

- Image component -> Optimize -> 40% images
- SEO extra features

Crawler

Nextjs

HTML

Server side rendered website

html

Nextjs

req

res

js + html

Data   React

website.com

html

Data   React

Next setup

Create a New Next.js Project:

- Open your terminal or command prompt.
- Navigate to the desired directory where you want to create the project.
- Run the following command:

```
→  Lecture_1 git:(main) ✗ npx create-next-app@latest next_setup
✔ Would you like to use TypeScript? … No / Yes
✔ Would you like to use ESLint? … No / Yes
✔ Would you like to use Tailwind CSS? … No / Yes
✔ Would you like to use `src/` directory? … No / Yes
✔ Would you like to use App Router? (recommended) … No / Yes
? Would you like to customize the default import alias (@/*)? › No / Yes
```
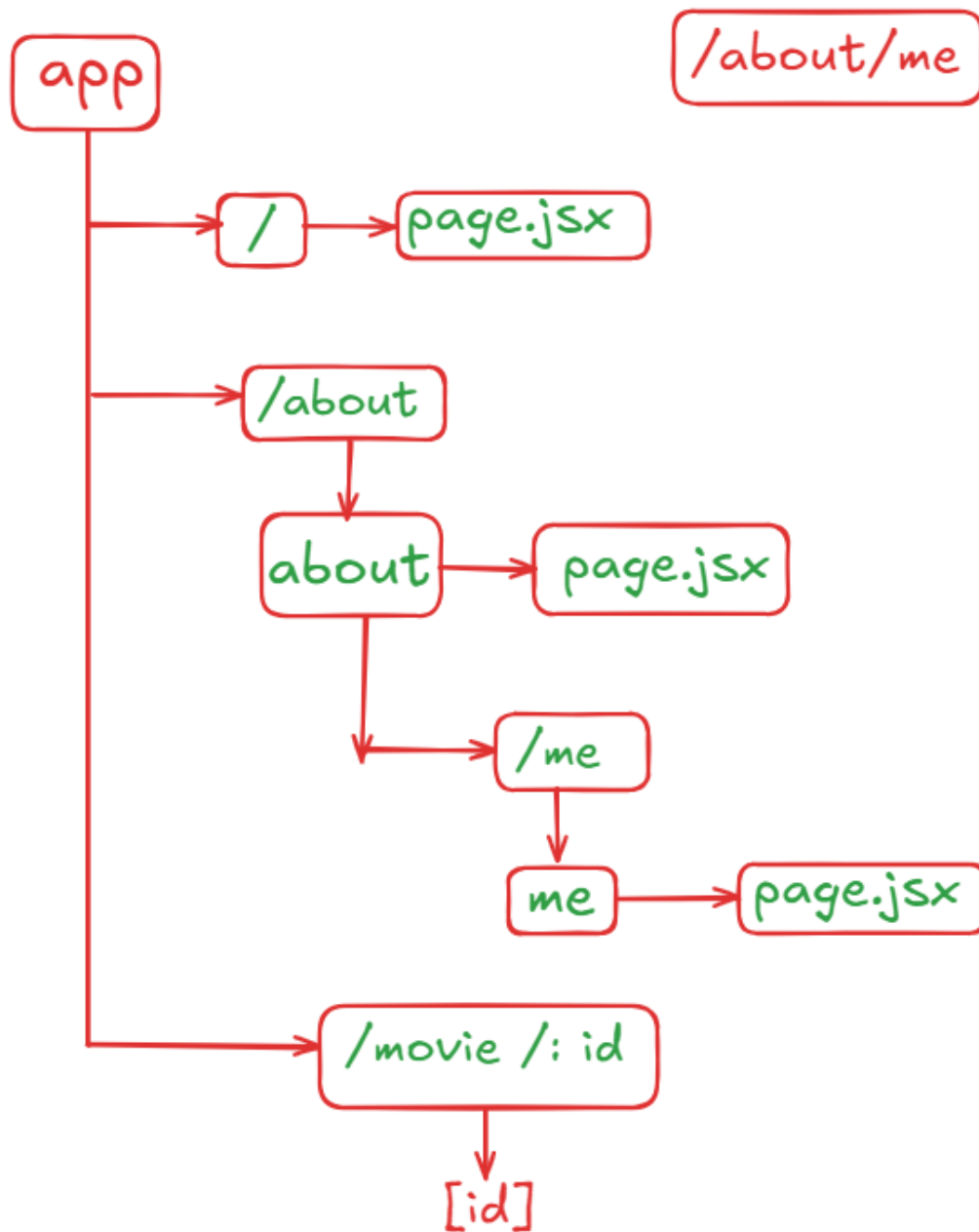
Navigate to your project directory:

```
cd next_setup
```

Run the following command:

```
npm run dev
```

This will start the development server, and you can access your Next.js application at `http://localhost:3000`.

Next Concepts

Routing (folder based)

In app directory this is `page.jsx`.

```jsx
import Image from "next/image";
import Link from "next/link";

export default function Home() {
  return (
    <div className="grid grid-rows-[20px_1fr_20px] items-center justify-items-center min-h-screen p-8
    pb-20 gap-16 sm:p-20 font-[family-name:var(--font-geist-sans)]">
      <main className="flex flex-col gap-8 row-start-2 items-center sm:items-start">
        I am home
      </main>
```

```
    </div>
  );
}
```

In `app` directory create a new directory name `about`.
In `about` directory this is `page.jsx`.

```
import React from 'react'

function AboutPage() {
  return (
    <div>About Page</div>
  )
}

export default AboutPage;
```

In `about` directory create a new directory name `me`.
In `me` directory this is `page.jsx`.

```
import React from 'react'

function Me() {
  return (
    <div>I am Me page</div>
  )
}

export default Me;
```

If you want to access `me` page then you have to write `/about/me` at the server.

Link

Link with href you will get from nextjs

```
<Link href="/about">About</Link>
<Link href="/contact">Contact</Link>
<Link href="/about/me">ME</Link>
```

Understanding Dynamic Routing

Dynamic routing in Next.js allows you to create routes that accept parameters, making your application more flexible and adaptable. By using brackets (`[slug]`) in your route paths, you can dynamically generate pages based on the values passed in the URL.

Define the Route: Create a JavaScript or TypeScript file in the `pages` directory with the dynamic route path. For example, `pages/[id].js` will handle routes like `/123`, `/456`, etc.

```jsx
import React from 'react'

function MovieData(props) {
        const id = props.params.id
  return (
    <div>Movie Data page for {id}</div>
  )
}

export default MovieData;
```
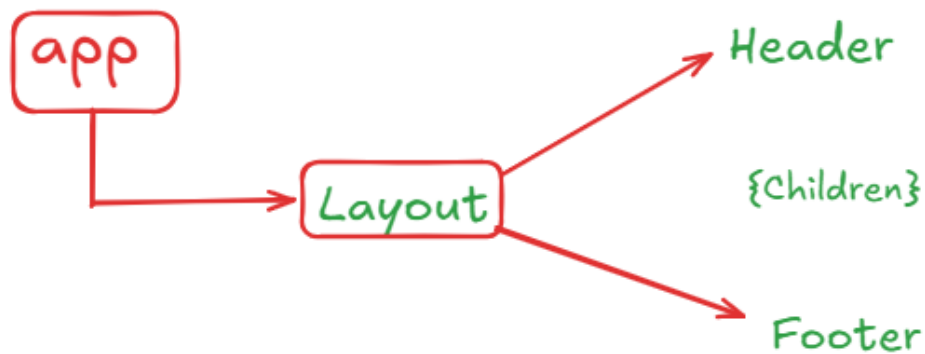
Understanding `layout.jsx`

- `layout.jsx` is a special file within the `pages` directory that can be used to wrap common components around other pages.
- By placing components in `layout.jsx`, you can ensure that they are rendered on every page that matches the corresponding route.

```jsx
import Head from 'next/head';

export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <body
        className={`${geistSans.variable} ${geistMono.variable} antialiased`}
      >
        <header>
          <h1 className="mx-auto">Header</h1>
        </header>
        {children}
        <footer>
          <h1 className="mx-auto">Footer</h1>
        </footer>
      </body>
```

```
        </html>
    );
}
```


Footer Component

Always create component directory out of app directory so whenever you want to access
you can easily access it.
This is `footer.jsx` in component directory.

```jsx
import React from 'react'

function Footer() {
    return (
        <footer>
            <p>---------------------------------</p>
            <h3>JioCinema</h3>
            <ul>
                <li>For You</li>
                <li>Sports</li>
                <li>Movies</li>
                <li>TV Shows</li>
            </ul>
        </footer>
    )
}

export default Footer
```

Now go to your `layout.jsx` and access this footer.

In `movie` directory `[movieId]` is sub directory.
In `[movieId]` this is `page.jsx`.

```jsx
import React from 'react'

async function fetchMovie(id) {
    const response = await
fetch(`https://freetestapi.com/api/v1/movies/${id}`);
    const data = await response.json();
    return data;
}

async function MovieData(props) {
    const movieId = props.params.movieId;
    const movieData = await fetchMovie(movieId);
    return (
        <>
            <p>----------------------</p>
            <h1>Movie Data Page for :  {movieId}</h1>
            <h2>Movie Name : {movieData.title}</h2>
            <p> Language Name : {movieData.language}</p>
        </>
    )
}

export default MovieData;
```

you can also get data from other `api` in `next js` and can send actual filled html to the frontend.