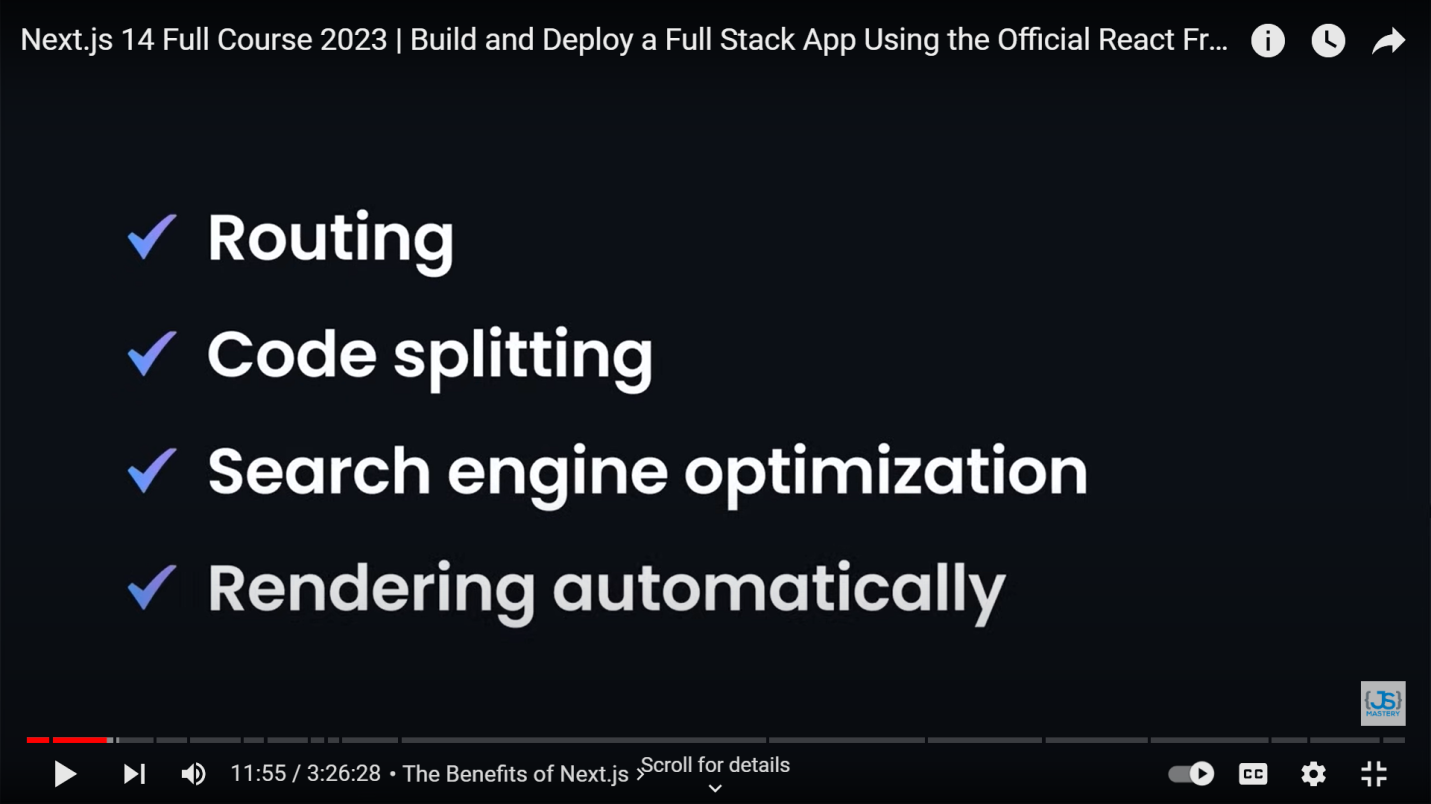
Rendering is React's process of describing a user interface based on the application's current state and props. The initial render in a React app is the first render when the application starts up, while re-rendering occurs when there is a change in the state to figure out which parts of the UI need an update.

* React js renders in client side, but Node JS performs rendering in server side, and also it is possible to choose where do we want it to get rendered.
* Use of Next JS helps SEO – Search Engine Optimization(it renders the page already in the server side and then send the rendered page to the client side whereas in react after the page request the server side sends the html,js,css pages to the client side and then it gets rendered on the client side.)
* Serverless APIs can be created in Next JS



**NEXT JS (Look Before Interview)**

1. Any changes made to **layout.js**, it will be displayed on every route page.
2. **Page.js –** is the home page. //localhost:3000/
3. If we are using any of the react hooks(ex-useState), we need to render the page on the client side, coz this takes place on client side. // ‘use client’;
4. **Routing** – to enable routing, create a folder in directory app.

**Data Fetching in Next.js**

1. SSR – server side rendering // within fetch({cache: ‘no-store’})
2. SSG – static site generation //default, it caches the data and stores it
3. ISR – Incremental static generation //within fetch({next :{revalidate:10}}). This method caches the data, but for the given time interval and again refreshes it. Hence, this is the best method to use.

For keeping backend, its advisable to make another folder under app, named api. So that we can understand that its is our backend and all other thing in app is frontend.

*===================================================================////////Now we are ready to get started with Full Stack Project in Next JS////////  
===================================================================*

1. Download mongoDB and mongoose other than next js packages.  
   npm i bcrypt mongodb mongoose next-auth  
   //*bcrypt – for encrypting passwords*
2. Delete app. Create folders app, components, utils, styles and models(for mongoose db models) and file .env(for storing keys).
3. As soon as u have made a setup for getProviders, create api/auth/[…neaxtauth]/route.js under app folder.
4. Create utils/database.js file, which will be used to connect to a database.  
   Make changes to Route.js also.