











## Section 8: Adding the client side application



0 / 22 | 3hr 7min

- ☐ 82. Introduction to Section 8  
 9min
- ☐ 83. Creating the NextJS project  
 8min
- ☐ 84. Reviewing and simplifying the NextJS Project  
 11min
- ☐ 85. Creating a nav bar  
 15min
- ☐ 86. Fetching data from the API  
 6min
- ☐ 87. Adding an Auction Card component  
 5min
- ☐ 88. Styling the auction cards  
 13min
- ☐ 89. Adding a countdown timer to the auction card  
 13min
- ☐ 90. Adding loading to the images  
 7min
- ☐ 91. Adding types to the project  
 6min
- ☐ 92. Adding pagination to our list  
 9min
- ☐ 93. Using server functions in client components  
 11min

- ☐ 94. Adding a set page size option  
 9min
- ☐ 95. Using Zustand for state management  
 10min
- ☐ 96. Refactoring our code to use the zustand state  
 11min
- ☐ 97. Adding a search bar  
 6min
- ☐ 98. Adding the search functionality  
 8min
- ☐ 99. Adding a reset to the search function  
 4min
- ☐ 100. Adding the sorting functionality  
 9min
- ☐ 101. Adding the filtering functionality  
 4min
- ☐ 102. Adding a component to display when zero results  
 11min
- ☐ 103. Summary of section 8  
 1min

# Why NextJS?

---

- Performance is excellent
- Load times helped with lazy loading and pre-fetching
- Good SEO due to Server side rendering
- Can act as BFF for our client app
- Can keep a secret
- React based



# NextJS 'Cons'

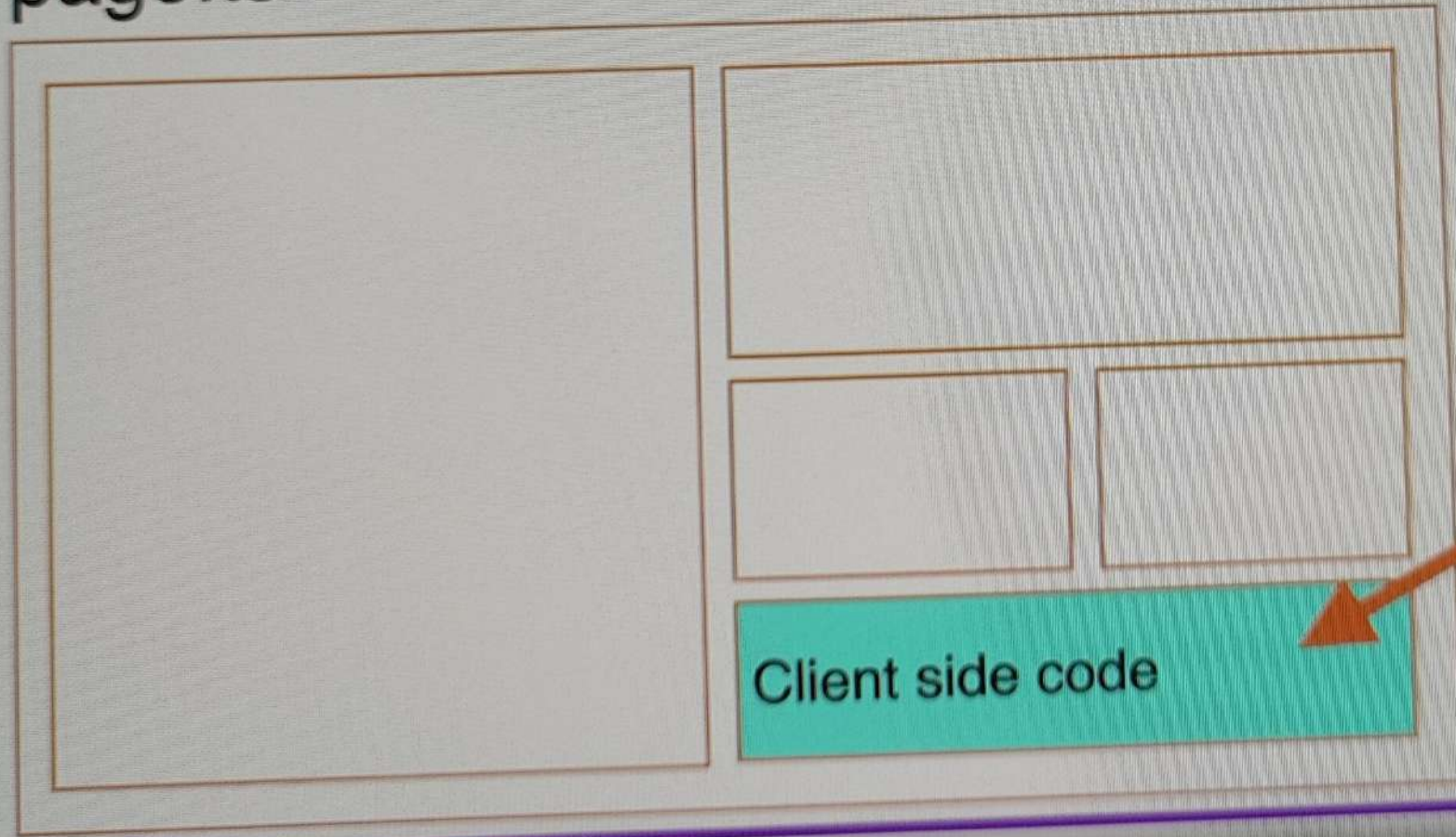
---

- Opinionated
- Some do not like how routing is handled
- React based



# Server and client side

page.tsx



Hydration

Client side code

# Expectations

We will not be creating a responsive app

We will not be creating a native mobile app

We will create a desktop web app



## In this section

---

- Created the next.js app
- Now have the read side of UI
- State management
- Server side actions