Interview Prep

I. Self Intro
   A. Name.
   B. Designation with exp level.
   C. Company.
   D. Tech skills.
   E. Project
   F. Reason for leaving and applying (if asked)

II. Projects
   A. Lionize.ai - (Babji, Deepika, Sruthi, Padmavathi, Chandana, Madhu)
      1. A project for influencer marketing (brands and creators/influencers)
      2. Developing UI to code from figma wireframes
      3. Frontend - React with typescript, zustand, tailwind
      4. Backend - Node, Express, MongoDB
   B. TeamNation sports - (Sai Eswar, Rakesh, Santhosh, Satwik, Kiran, Bhavani)
      1. Project related to american football
      2. Web and Mobile app for Players and coaches
      3. Frontend - React, typescript, zustand, material ui, ionic-react for mobile.
      4. Backend - Node, Express, MongoDB
   C. Energyfunders - (venkat sai, Rohan, Sagar, Kumar, Subash, Kavya)
      1. Project related to Oil and gas investment platform.
      2. Frontend - React with typescript, zustand, tailwind
      3. Backend - Node, Express, MongoDB
      4. Complex task - Dwolla js payment integration. (Mass payments)
   D. Dreamhyre (Optional)
      1. Landing Page for Dreamhyre
      2. Frontend React, tailwind
   E. POC's -
      1. Influenza
      2. Zap

III. Technical
   A. Frontend

1. **Technical terms** - Clients/Users, elements, DOM (tree), specificity, component, state, prop, callback as a function, global state, hooks, lifecycle, modularization, Virtual DOM, Rendering, routing.
2. HTML
    a) Version
        (1) HTML5
        (2) Features
            (a) Several new features were introduced in HTML5 like semantics…etc
    b) What is Meta tag
        (1) Meta tag is used to provide extra info to the website like SEO optimization, Mobile device optimization, Analytics.
    c) Difference btw div tag and span tag.
        (1) Div tag is as block level element which is used as a container while span tag is an inline element used for applying styles or scripting to a specific portion of text within a larger block-level element.
    d) Attributes-
    e) Lists
        (1) Ordered List - Ordered by numbers or other characters.
        (2) Unordered List - Represented by Dot notation.
    f) Table Tags
    g) Fav icon (Used to set an icon to be displayed on the browser tab)
        (1) <link rel="icon" href="/favicon.ico" type="image/x-icon">
    h) Form - (Used to get info from the user/client)
        (1) Attributes - method, action, onsubmit .
        (2) Every form needs on submit button (using type=”submit”)
3. CSS
    a) Id - unique identification given to a single element.
    b) Class - common identification given to multiple elements.

c) Box model - Each element/content in the DOM is wrapped in a box like model, which consists of content -> padding -> border -> margin
d) Flexbox - CSS layout module that provides a flexible way to create responsive and dynamic layouts for webpages.
    (1) Display: flex
    (2) Flex-direction
    (3) Justify-content
    (4) Align-items
    (5) Flex-wrap
    (6) Gap
    (7) Flex
e) Center a div - We can center a div using either a flexbox or grid.
f) Display:none - It completely removes the element from the DOM (both element and the space)
g) Visibility:hidden - It removes/hides the element from the DOM but the space is still taken.
h) Overflow2
i) Z-index
j) Position.
k) margin/padding.
l) border/border-radius.
m) Opacity.
n) transform/transition
o) Box-shadow
p) Pseudo classes/elements
q) Media queries - Queries to change styling of elements based on width/height at that particular time.

4. Javascript
    a) **Datatypes**- primitive & non-primitive
    b) Variables
        (1) Const-variables with a constant value
        (2) Let-allows reassignment
        (3) Var-redeclared and updated

c) **Hoisting** - all the variable declarations are moved at the top of the scope before code execution.
d) **Closure** (ex - debounce)
e) **Abstraction**-An abstraction is a way of hiding the implementation details and showing only the functionality to the users., inheritance-reuse properties and methods of an existing class when you create a new class, polymorphism- a core concept of an object-oriented paradigm that provides a way to perform a single action in different forms, encapsulation
f) Class
g) **JSON methods**-(json.parse & json.stringify)
h) String methods-
i) Array methods-
j) Object methods-
k) Event loop
l) **Callback hell** - Nested callback functions.
m) **Promises** -
    (1) Promises are used in asynchronous programming.
    (2) We declare a promise which has resolve and reject, resolve is used to fulfill the is resolved or rejected it is in pending state.promise, reject is for rejecting the promise. Until it
    (3) Promise.all - Used to call multiple promises at once and get the response only if all the promises are fulfilled.
n) **Async/await** - New feature introduced in es6, we can replace promises with async/await. Used to write asynchronous programming.
o) **Call**-is a function that helps you change the context of the invoking function. In layperson's terms, it helps
p) you replace the value of this inside a function with whatever value you want.

Apply- is very similar to the `call` function. The only difference is that in apply you can pass an array as an argument list.

Bind-is a function that helps you create another function that you can execute later with the new context of this that is provided.

q) **Try catch** - used for error handling, We wrap our logic in try and if there are any unintentional errors we catch them using catch.
r) Throwl
s) Cookie, cache, localstorage, sessionstorage
t) settimeout/setinterval
u) **Debounce** - It is a function used to avoid multiple api calls being called when the user clicks the submit button multiple times.
v) Events
w) **Pure Function** - A function which returns the same output no matter how many times the same input is given.
x) **Higher order function** - Whenever we need some information to be passed to a child function, We wrap it with a higher order function.
y) Event delegation
z) Immutability
aa) Best practices -
    (1) Naming convention.
    (2) Comments.
    (3) Formatting.
    (4) Strict equality.
    (5) Break down into small modules/functions.
bb) Spread/Rest operator

5. React
    a) What is React?
        (1) React is a JS frontend library.
    b) Why React?
        (1) Virtual DOM -

(a) The virtual DOM is a clone of real DOM. React generates a virtual DOM representation of components and compares it with the previous version to determine minimal changes. It then efficiently updates only the necessary parts of the actual DOM, resulting in improved performance.

c) Lifecycle
   (1) Mounting
   (2) Updating
   (3) Unmounting
d) useEffect -
   (1) [] -> mounting
   (2) [stateVar] -> triggered when the stateVar is updated.
   (3) No second argument -> Called when any of the state variables or props are changed.
e) State, Props, Callback as a prop, Global state
f) **Redux**: UI -> actions -> reducers -> store
g) **Hooks** - useState, useEffect, useRef, useNavigate, useParams, useMemo, useCallback, useLayoutEffect
h) Prop-drilling

IV. Work experience -
   A. Company - Dreamhyre
      1. Project - Lionize
         a) A project for influencer marketing (brands and creators/influencers)
         b) Developing UI to code from figma wireframes
         c) Frontend - React with , zustand, tailwind
         d) Backend - Node, Express, MongoDB

V. Github (Version management) -
   A. Github is for version control.
   B. Git is not Github.
   C. Terminology:
      1. Checkout - To switch branch
         a) Git checkout develop

2. Pull - to pull latest code from a remote branch.
    a) Git pull - get all updates from all branches
    b) Git pull origin develop
3. Checkout with -b -> To create a new branch
    a) Git checkout -b loginPage-1736

4. Add - Add files to be committed
    a) Git add . (or) git add ./file.jsx
5. Commit - Add commit message before committing
    a) Git commit -m “commit message”
6. Rebase - Pull the latest code and add my code on top of it.
    a) Git pull –rebase origin develop
7. Push - to push your local code to a remote branch.
    a) Git push origin loginPage-1731
8. Log - see previous logs/commits
    a) Git log
9. Status - see current status
    a) Git status
10. Stash - Delete your changes till now but save them locally.
    a) Git stash (or) git stash –include-untracked
11. Stash pop - restore previously stashed code.
    a) Git stash pop
12. Reset - removing added files/reset Head commit
    a) Git reset
    b) Git reset –hard commitHash
13. Git merge - check it out