Redux & Router Questions

* **Whats were the major problems with MVC framework?** 
  + Some of the following are major problems with MVC framework

1. DOM manipulation was very expensive

2. Application were slow & inefficient

3. There was huge memory wastage

4. Because of circular dependencies, a complicated model was created around models & views

* **What is Flux?** 
  + Flux is an architectural pattern which enforces the uni-directional data flow
  + It controls derived data & enables communication between multiple components using a central store which has authority for all data
  + Any update in data throughout the application must occur here only, flux provides stability to the application & reduces run-time errors
* **What is Redux?** 
  + Its is a predictable state container for javascript applications & is used for the entire applications state management
  + Applications developed with redux are easy to test & can run in different environments showing consistent behavior
* **What do you understand by “Single source of truth”?** 
  + Redux uses ‘Store’ for storing the application’s entire state at one place
  + So all the component’s state are stored in the Store & they receive updates from the Store itself.
  + The single state tree makes it easier to keep track of changes over time & debug or inspect the application
* **What are the three principles that Redux follows?** 
  + **Single source of truth**, the state of the entire application is stored in an object/state tree within a single store
  + **State is read-only**, the only way to change the state is to trigger an action. An action is a plain JS object describing the change. Just like state in the minimal representation of data, the action is the minimal representation of the change to that data
  + **Changes are mode with pure functions**, in order to specify how the state tree is transformed by actions, you need pure functions. Pure function are those whose return value depends solely on the values of their arguments
* **List down the components of Redux?** 
  + Redux is composed of the following components

1. **Action**, its an object that describes what happened

2. **Reducer**, it is place to determine how the state will change

3. **Store**, state/object tree of the entire application is saved in the store

4. **View**, simply display the data provided by the store

* **How does data flow through redux?** 
  + An action is dispatched when a user interacts with the application
  + The root reducer functions is called with the current state & the dispatched action. The root reducer may divide the task among smaller reducer, which ultimately returns a new state
  + The store notifies the view by executing their callback functions
  + The view can retrieve updated state & re-render again
* **How are Actions defined in Redux?** 
  + Actions in React must have a type property that indicates the type of ACTION being performed
  + They must defined as a String constant & you can add more properties to it as well.
  + In Redux, actions are created using the function called Action Creator
* **What is the role of the Reducer?** 
  + Reducer are pure functions which specify how the applications state changes in response to an ACTION. Reducers work by taking in the previous state & action, and then it returns a new state.
  + It determines what sort of update needs to be done based on the type of the action, and then returns new values. It returns the previous state as it is, if no work needs to be done
* **What is the significance of Store in Redux?** 
  + A store is javascript object which can hold the applications state & provide a few helper methods to access the state, dispatch actions & registers listeners
  + The entire state/object is saved in a single store, as result of this Redux is very simple & predictable
  + We can pass middleware to the store to handle the processing of data as well as to keep a log of various action that change the state of stores all actions return a new state via reducers
* **How is Redux different from Flux?** 
  + **Flux**, the store contains state & change logic, there are multiple stores, all the stores are disconnected & flat, has singleton dispatcher, React components subscribe to the store, & state is mutable
  + **Redux**, store & change logic are separated, there is only one store, single store with hierarchical reducers, no concept of dispatcher, containers components utilize connect & state is immutable
* **Why do we need a Router in React?** 
  + Router is used to define multiple routes & when a user types a specific URL, if this URL matches the path of any ‘route’ defined inside the router, then the user is redirected to the particular route
* **How are the advantages of Redux?** 
  + Advantage of Redux are listen below

1. **Predictability of outcome**, since there is always one source of truth, i.e. the store there is no confusion about how to sync the current state w/ actions & other parts of the application

2. **Maintainability**, the code becomes easier to maintain with a predictable outcome & strict structure

3. **Server-side rendering**, you just need to pass the store created on the server, to the client side. This is very useful for initial render

4. **Developer tools**, from actions to state change, developers can track everything going on in the application in real time

5. **Community & ecosystem**, Redux has a huge community behind it which makes it even more captivating to use.

6. **Ease of testing**, Redux code is mostly functions which are small, pure & isolated. This makes the code testable & independent

7.**Organization**, Redux is precise about how code should be organized, this makes the code more consistent & easier when a team work with it

* **What is React Router?** 
  + React Router is powerful routing library built on top of React, which helps in adding new screens & flows to the application.
  + This keeps the URL in sync w/ data that’s being displayed on the web page.
  + It maintains a standardized structure & behavior & is used for developer single page web applications
* **Why is switch keyword used in React Router v4?** 
  + Although a <div> is used to encapsulate multiple routes inside the Router, the ‘switch’ keyword is used when you want to display only a single route to be rendered amongst the several defined routes
  + The <switch> tag when in use matches the type URL with the defined routes in sequential order
  + When the first match is found, it renders the specified router thereby bypassing the remaining route