DAY 26: MORNING ASSESSMENT

Unit Testing ReactJS Apps (Jest & Enzyme)

Q1. What is the main purpose of Jest in React projects?

a) State management

b) Unit testing framework

c) CSS styling tool

d) Routing library

option: b

answer: Unit testing framework

Q2. Which command is commonly used to run Jest tests?

a) npm run start

b) npm run test

c) npm run build

d) npm run lint

option: b

answer: npm run test

Q3. Enzyme is primarily used for:

a) API requests

b) Component testing

c) Redux store management

d) Internationalization

option: b

answer: Component testing

Q4. Which Jest function is used to group related test cases?

a) test()

b) expect()

c) describe()

d) it()

option: c

answer: describe()

Q5. In Jest, which function is used to make assertions?

a) test()

b) expect()

c) render()

d) shallow()

option: b

answer: expect()

Q6. What does the shallow() method from Enzyme do?

a) Fully renders the component tree

b) Renders only the given component without its children

c) Tests API endpoints

d) Creates Redux store mocks

option: b

answer: Renders only the given component without its children

Q7. Which function is used to render a component with full DOM in Enzyme?

a) shallow()

b) render()

c) mount()

d) snapshot()

option: c

answer: mount()

Q8. Which Jest matcher is used to check strict equality?

a) toBe()

b) toEqual()

c) toMatch()

d) toContain()

option: a

answer: toBe()

Q9. What does Jest's toEqual() check for?

a) Reference equality

b) Deep equality of objects/arrays

c) String matching

d) Boolean comparison

option: b

answer: Deep equality of objects/arrays

Q10. Which Enzyme method simulates user interactions?

a) simulate()

b) trigger()

c) act()

d) mockEvent()

option: a

answer: simulate()

Q11. Jest's beforeEach() function is used to:

a) Run once before all tests

b) Run before every test case

c) Run once after all tests

d) Run only if tests fail

option: b

answer: Run before every test case

Q12. Snapshot testing in Jest ensures that:

a) API responses match schema

b) Component output hasn’t unexpectedly changed

c) Redux store is updated

d) Database schema is consistent

option: b

answer: Component output hasn’t unexpectedly changed

Q13. Which file extension is commonly used for Jest test files?

a) .jsx

b) .test.js

c) .snap.js

d) .mock.js

option: b

answer: .test.js

Q14. Which Jest function is used to mock dependencies?

a) jest.fn()

b) jest.mock()

c) jest.spyOn()

d) All of the above

option: d

answer: All of the above

Q15. What does enzyme-adapter-react-16 (or latest) do?

a) Adds Redux support

b) Bridges Enzyme with React version

c) Enables Jest matchers

d) Creates snapshots

option: b

answer: Bridges Enzyme with React version

Q16. In Jest, which function is used to run a single test file?

a) npm test fileName.test.js

b) jest --runSingle

c) npm run build fileName

d) testOnly fileName

option: a

answer: npm test fileName.test.js

Q17. Enzyme’s find() method is used to:

a) Locate a DOM node or component

b) Simulate user clicks

c) Create a snapshot

d) Mount the component

option: a

answer: Locate a DOM node or component

Q18. Which Jest configuration file is commonly used?

a) jest.config.json

b) jestfile.json

c) test.config.js

d) enzyme.config.js

option: a

answer: jest.config.json

Q19. Which Jest function is used to check if a function is called?

a) toBeCalled()

b) toHaveBeenCalled()

c) wasCalled()

d) fnCalled()

option: b

answer: toHaveBeenCalled()

Q20. In Jest, which option enables coverage reports?

a) npm test --verbose

b) npm test --coverage

c) npm test --watch

d) npm test --report

option: b

answer: npm test --coverage

E2E Testing using Cypress

Q21. Cypress is mainly used for:

a) Backend testing

b) API testing

c) End-to-End testing

d) Database testing

option: c

answer: End-to-End testing

Q22. Cypress tests run directly inside the:

a) Browser

b) Server

c) Database

d) IDE

option: a

answer: Browser

Q23. Which Cypress command is used to visit a page?

a) cy.load()

b) cy.goto()

c) cy.visit()

d) cy.open()

option: c

answer: cy.visit()

Q24. Cypress test files are usually written in:

a) HTML

b) JavaScript/TypeScript

c) Python

d) C#

option: b

answer: JavaScript/TypeScript

Q25. Cypress default folder for integration tests is:

a) /cypress/tests/

b) /cypress/integration/

c) /tests/integration/

d) /src/tests/

option: b

answer: /cypress/integration/

Q26. Cypress supports which type of testing?

a) Unit

b) Component

c) Integration

d) All of the above

option: d

answer: All of the above

Q27. To get an element by ID in Cypress:

a) cy.getElement('#id')

b) cy.find('#id')

c) cy.get('#id')

d) cy.locate('#id')

option: c

answer: cy.get('#id')

Q28. Which Cypress command is used to simulate a click?

a) cy.press()

b) cy.tap()

c) cy.click()

d) cy.triggerClick()

option: c

answer: cy.click()

Q29. Cypress provides:

a) Real browser environment

b) Virtual DOM only

c) API mocks only

d) Only CLI support

option: a

answer: Real browser environment

Q30. Cypress runs asynchronously but provides:

a) Callbacks

b) Automatic waiting

c) Promises only

d) Manual waits

option: b

answer: Automatic waiting

Q31. Cypress command to type text into an input:

a) cy.enter('text')

b) cy.type('text')

c) cy.write('text')

d) cy.input('text')

option: b

answer: cy.type('text')

Q32. Cypress command to assert visibility:

a) cy.get().visible()

b) cy.should('be.visible')

c) cy.expect('visible')

d) cy.assert('visible')

option: b

answer: cy.should('be.visible')

Q33. Cypress supports test retries with:

a) Retry block

b) Cypress.config({ retries })

c) cy.retry()

d) cy.loop()

option: b

answer: Cypress.config({ retries })

Q34. Cypress dashboard is used for:

a) Real-time test results & analytics

b) Code compilation

c) Storing test code

d) Database queries

option: a

answer: Real-time test results & analytics

Q35. Cypress automatically handles:

a) Async code & waits

b) SQL queries

c) Cloud deployments

d) Webpack builds

option: a

answer: Async code & waits

Q36. In Redux, the state must always be treated as:

a) Mutable

b) Immutable

c) Temporary

d) None of the above

option: b

answer: Immutable

Q37. Which function in Redux is used to combine multiple reducers into one?

a) createReducer

b) applyMiddleware

c) combineReducers

d) rootReducer

option: c

answer: combineReducers

Q38. Which Redux middleware is commonly used for handling asynchronous operations?

a) redux-logger

b) redux-thunk

c) redux-toolkit

d) redux-persist

option: b

answer: redux-thunk

Q39. In Redux, what does dispatch() do?

a) Updates the reducer directly

b) Sends an action to the store

c) Returns the current state

d) Initializes the store

option: b

answer: Sends an action to the store

Q40. What is the correct order of Redux data flow?

a) Store → Reducer → Action → UI

b) UI → Action → Reducer → Store → UI

c) Reducer → Store → Action → UI

d) Action → UI → Reducer → Store

option: b

answer: UI → Action → Reducer → Store → UI

Q41. When fetching data in Redux, where should the API call usually be placed?

a) Inside the reducer

b) Inside the action creator (with middleware)

c) Inside the store directly

d) Inside the component only

option: b

answer: Inside the action creator (with middleware)

Q42. Which hook is often used in React-Redux for accessing state from the store?

a) useEffect

b) useReducer

c) useSelector

d) useContext

option: c

answer: useSelector

Q43. Which hook is used in React-Redux to dispatch actions?

a) useDispatch

b) useSelector

c) useReducer

d) useAction

option: a

answer: useDispatch

Q44. What does an action in Redux contain?

a) Reducers and state

b) State and middleware

c) Type and payload

d) Store and UI

option: c

answer: Type and payload

Q45. What happens if you try to mutate Redux state directly?

a) Redux automatically corrects it

b) State updates without problems

c) It breaks time-travel debugging and immutability principles

d) Nothing happens

option: c

answer: It breaks time-travel debugging and immutability principles

Q46. Which of the following is NOT true about Redux store?

a) It holds the application state

b) It allows direct modification of state

c) It is updated only through dispatching actions

d) It is created using createStore or configureStore

option: b

answer: It allows direct modification of state

Q47. What is the main purpose of Redux middleware?

a) To directly update the reducer

b) To log only errors

c) To intercept actions before they reach the reducer

d) To remove boilerplate code

option: c

answer: To intercept actions before they reach the reducer

Q48. Which library is recommended by the Redux team for writing Redux logic?

a) redux-observables

b) redux-toolkit

c) redux-thunk

d) redux-persist

option: b

answer: redux-toolkit

Q49. What does the mapStateToProps function do in React-Redux?

a) Maps component props to reducer

b) Maps Redux state to React component props

c) Maps React component props to Redux store

d) Maps dispatch to reducers

option: b

answer: Maps Redux state to React component props

Q50. If you want to initialize state in Redux with server data, which lifecycle stage is best?

a) Before rendering the component (useEffect)

b) Inside the reducer directly

c) Only during store creation

d) Inside UI event handlers

option: a

answer: Before rendering the component (useEffect)

RxJS & Redux-Observables, Reducers & Actions

Q51. What does RxJS primarily help with in React applications?

a) Managing UI components

b) Handling asynchronous data streams

c) Styling components

d) Building server-side apps

option: b

answer: Handling asynchronous data streams

Q52. What is a Redux-Observable?

a) A middleware for handling async logic with RxJS

b) A library for styling React apps

c) A testing framework for React

d) A server-side rendering library

option: a

answer: A middleware for handling async logic with RxJS

Q53. Which RxJS operator is commonly used in Redux-Observable epics for mapping actions?

a) switchMap

b) mapState

c) reduce

d) dispatchMap

option: a

answer: switchMap

Q54. In Redux, what is the role of a reducer?

a) Dispatching actions

b) Describing state changes based on actions

c) Fetching data from API

d) Handling authentication

option: b

answer: Describing state changes based on actions

Q55. Which of the following best describes Redux actions?

a) Functions that modify state directly

b) Pure functions

c) Plain JavaScript objects describing changes

d) Observables that emit state

option: c

answer: Plain JavaScript objects describing changes

Q56. Which operator cancels the previous observable when a new one is emitted?

a) mergeMap

b) switchMap

c) concatMap

d) map

option: b

answer: switchMap

Q57. What is the purpose of implementing Undo History in Redux?

a) To support rollback of UI state

b) To manage API requests

c) To improve performance

d) To simplify reducers

option: a

answer: To support rollback of UI state

Q58. Which Redux concept is most important for supporting undo/redo?

a) Middleware

b) Reducer composition

c) Immutable state

d) Async actions

option: c

answer: Immutable state

Q59. What does ImmutableJS provide?

a) Fast rendering

b) Persistent, immutable data structures

c) Authentication management

d) A testing framework

option: b

answer: Persistent, immutable data structures

Q60. Which of the following is a key advantage of ImmutableJS in Redux apps?

a) Larger bundle size

b) Direct state mutation

c) Performance optimization with structural sharing

d) Automatic API fetching

option: c

answer: Performance optimization with structural sharing

Q61. In RxJS, which operator is best for handling multiple values sequentially without cancellation?

a) switchMap

b) concatMap

c) mergeMap

d) flatMap

option: b

answer: concatMap

Q62. Which function is used in Redux to combine multiple reducers?

a) mergeReducers()

b) joinReducers()

c) combineReducers()

d) useReducers()

option: c

answer: combineReducers()

Q63. In Redux-Observable, what is an Epic?

a) A reducer that handles async data

b) A middleware function that listens for actions and returns new actions

c) A UI component that renders state

d) A Redux store enhancer

option: b

answer: A middleware function that listens for actions and returns new actions

Q64. ImmutableJS provides which method to update deeply nested data without mutation?

a) setDeep()

b) updateIn()

c) pushState()

d) changeIn()

option: b

answer: updateIn()

Q65. In Undo History implementation, which Redux principle ensures previous states can be restored?

a) Single source of truth

b) Actions must be pure

c) State is immutable

d) Store is asynchronous

option: c

answer: State is immutable

Redux-Thunk & Redux-Saga MCQs

66. What is the primary purpose of Redux-Thunk?

a) To handle routing in React apps

b) To write action creators that return functions instead of actions

c) To test Redux reducers

d) To optimize React rendering

option: b

Answer: To write action creators that return functions instead of actions

67. In Redux-Thunk, the function returned by an action creator receives:

a) Only dispatch

b) Only getState

c) Both dispatch and getState

d) Neither

option: c

Answer: Both dispatch and getState

68. Which of the following is a use case for Redux-Thunk?

a) Testing React components

b) Managing asynchronous API calls

c) Writing reducers

d) Handling routing

option:b

Answer: Managing asynchronous API calls

69. Redux-Saga is based on which JavaScript feature?

a) Promises

b) Generators

c) Callbacks

d) Async/Await

option: b

Answer : Generators

70. In Redux-Saga, which effect is used to call asynchronous functions?

a) put

b) call

c) take

d) delay

option: b

Answer: call

71. Which Redux middleware is best for complex async workflows like cancellation and sequencing?

a) Redux-Thunk

b) Redux-Saga

c) Redux-Logger

d) Redux-Observable

option: b

Answer: Redux-Saga

72. The put effect in Redux-Saga is used to:

a) Dispatch an action

b) Call an API

c) Cancel a task

d) Create a reducer

option: a

Answer: Dispatch an action

73. In Redux-Saga, the takeEvery effect does what?

a) Cancels previous sagas

b) Runs a saga for every matched action

c) Runs only the latest action saga

d) Runs sagas sequentially

option:b

Answer: Runs a saga for every matched action

74. Which effect in Redux-Saga runs only the latest task and cancels previous ones?

a) takeLatest

b) takeEvery

c) fork

d) join

option:a

Answer :takeLatest

75. What is a key difference between Redux-Thunk and Redux-Saga?

a) Thunk uses Promises, Saga uses Generators

b) Thunk is synchronous, Saga is asynchronous

c) Thunk is faster than Saga

d) Saga does not support async

option: a

Answer: Thunk uses Promises, Saga uses Generators

76. Which middleware allows you to retry failed API calls automatically?

a) Redux-Thunk

b) Redux-Saga

c) Redux-Persist

d) Redux-Logger

option: b

Answer:Redux-Saga

77. In Redux-Saga, yield call(apiFunction) ensures:

a) The function is executed asynchronously

b) The action is dispatched

c) The reducer is updated

d) The store is replaced

option: a

Answer: The function is executed asynchronously

78. Which of these is NOT true about Redux-Thunk?

a) It is simpler to learn than Redux-Saga

b) It uses functions instead of plain actions

c) It is good for small to medium apps

d) It requires generator functions

option:d

Answer: It requires generator functions

79. Which effect is used to pause execution in Redux-Saga for a given time?

a) wait

b) delay

c) timeout

d) sleep

option: b

Answer: delay

80. Which scenario would benefit more from Redux-Saga than Redux-Thunk?

a) Simple API calls

b) Complex async workflows with cancellation

c) Dispatching synchronous actions

d) Static state

option:b

Answer: Complex async workflows with cancellation

81. What is the main purpose of Redux-Thunk?  
a) Handle asynchronous logic in Redux  
b) Manage routing in Redux apps  
c) Optimize rendering performance  
d) Create reducers automatically  
option: a  
answer: Handle asynchronous logic in Redux

82. Redux-Thunk allows dispatching of:  
a) Only objects  
b) Functions and objects  
c) Only strings  
d) Reducers directly  
option: b  
answer: Functions and objects

83. Which middleware is required to enable Redux-Thunk?  
a) redux-saga  
b) redux-devtools  
c) redux-thunk  
d) redux-persist  
option: c  
answer: redux-thunk

84. In Redux-Saga, which effect is used to call an asynchronous function?  
a) take  
b) call  
c) put  
d) select  
option: b  
answer: call

85. What does the put effect in Redux-Saga do?  
a) Calls an API  
b) Dispatches an action  
c) Waits for an action  
d) Cancels a task  
option: b  
answer: Dispatches an action

86. Redux-Saga is built on top of:  
a) Async/Await  
b) Observables  
c) Generator functions  
d) Promises only  
option: c  
answer: Generator functions

87. In Redux-Saga, the takeLatest effect is used to:  
a) Run all actions in sequence  
b) Cancel previous tasks and run only the latest one  
c) Queue all actions without canceling  
d) Run actions in parallel  
option: b  
answer: Cancel previous tasks and run only the latest one

88. Redux-Thunk is best suited for:  
a) Complex async flows with cancellation  
b) Simple async logic like API calls  
c) State immutability handling  
d) Reducer composition  
option: b  
answer: Simple async logic like API calls

89. Which effect in Redux-Saga is used to watch for dispatched actions?  
a) take  
b) call  
c) race  
d) put  
option: a  
answer: take

90. Redux-Saga can handle:  
a) Only synchronous logic  
b) Complex asynchronous workflows  
c) CSS styling  
d) DOM rendering  
option: b  
answer: Complex asynchronous workflows

91. Which of the following is TRUE about Redux-Thunk?  
a) It uses generator functions  
b) It allows writing action creators that return functions  
c) It replaces reducers  
d) It is built on RxJS  
option: b  
answer: It allows writing action creators that return functions

92. In Redux-Saga, the select effect is used to:  
a) Fetch data from an API  
b) Access the Redux store state  
c) Dispatch an action  
d) Cancel an effect  
option: b  
answer: Access the Redux store state

93. Which is an advantage of Redux-Saga over Redux-Thunk?  
a) Simpler for beginners  
b) Handles side effects more declaratively  
c) No need for middleware  
d) Faster rendering always  
option: b  
answer: Handles side effects more declaratively

94. Which effect in Redux-Saga allows running multiple effects in parallel?  
a) race  
b) all  
c) call  
d) fork  
option: b  
answer: all

95. In Redux-Saga, what does the fork effect do?  
a) Run a blocking call  
b) Spawn a non-blocking task  
c) Cancel all tasks  
d) Dispatch a reducer  
option: b  
answer: Spawn a non-blocking task

Q96. What does i18n stand for in software development?

a) Integration

b) Internationalization

c) Interaction

d) Initialization

option: b

answer: Internationalization

Q97. How many letters are between the first and last letters in “Internationalization” that form i18n?

a) 15

b) 17

c) 18

d) 19

option: c

answer: 18

Q98. Which React library is most commonly used for i18n support?

a) react-intl

b) react-localize-redux

c) react-i18next

d) next-translate

option: c

answer: react-i18next

Q99. In react-i18next, what is the hook used to access translation functions?

a) useLocale

b) useLang

c) useI18n

d) useTranslation

option: d

answer: useTranslation

Q100. Which JSON structure is typically used for storing translations?

a) Array of strings

b) Key-value pairs

c) CSV format

d) XML format

option: b

answer: Key-value pairs

Q101. In i18n, what is “l10n”?

a) Localization

b) Lexicalization

c) Legalization

d) Language-switching

option: a

answer: Localization

Q102. Which of the following is NOT an i18n challenge?

a) Different date formats

b) Currency conversion

c) String interpolation

d) Component state management

option: d

answer: Component state management

Q103. In i18next, what option allows you to fallback to a default language?

a) defaultLang

b) fallbackLng

c) baseLocale

d) backupLang

option: b

answer: fallbackLng

Q104. Which of these locales is valid for US English?

a) en-uk

b) en-us

c) eng-us

d) en-english

option: b

answer: en-us

Q105. What is the purpose of ICU message formatting in i18n?

a) To format database queries

b) To support pluralization and gender rules

c) To compress translation files

d) To sort translation keys

option: b

answer: To support pluralization and gender rules

Q106. What does the Trans component in react-i18next help with?

a) Language detection

b) Rendering translated strings with React elements inside

c) Switching between locales

d) Storing translations in cookies

option: b

answer: Rendering translated strings with React elements inside

Q107. Which of the following is NOT a feature of react-i18next?

a) Lazy loading translations

b) Context-based translations

c) Automatic pluralization

d) Automatic code-splitting

option: d

answer: Automatic code-splitting

Q108. If a translation key is missing, what will i18next usually display?

a) Error message

b) Empty string

c) The key itself

d) Default English

option: c

answer: The key itself

Q109. Which React feature can help with dynamic text changes in multiple languages?

a) Context API

b) useMemo

c) useEffect

d) Suspense

option: a

answer: Context API

Q110. Which statement is TRUE about localization?

a) It always uses English as fallback

b) It adapts content to specific regions/cultures

c) It only changes the text language

d) It ignores formatting of dates and numbers

option: b

answer: It adapts content to specific regions/cultures