

***** Tweeter Starter Code *****

- Milestone 1

- Demo (login/register, toasts)

***** Let's review React concepts in the context of Tweeter client *****

CONCEPT: Web Page = HTML + Javascript + Events

- simple.html

CONCEPT: DOM (image)

- DOM.jpg

CONCEPT: Traditional Web App vs Single Page Application (SPA)

- User points browser at web app's URL

- Retrieves root page from server

CONCEPT: Web Client Frameworks

- Before web client frameworks came along, browser programming was a mess

 - Angular, Vue, React

- UI is decomposed into "components"

35 - Each component is responsible for
rendering part of the DOM tree, and
handling user events in its part of the
tree

36

37 - The root component renders the top part
of the DOM, and invokes other components
to render sub-parts of the DOM

38

39

40 CONCEPT: In React, components are functions
that render part of the DOM

41

42 - Create basic React app using Vite build
tool

43

44 - "Create React TypeScript App with Vite"
slide

45 - npm create vite@latest vite-app --
--template react-ts

46

47 - App component is function that returns
its part of the DOM, including event
handlers

48

49 - JSX is declarative description of the
component's DOM (declarative vs.
procedural)

50

51 - JSX compiler converts JSX code into
procedural JavaScript code that updates
the DOM

52

53 - Change App to use "let count = 0" and
"count + 1" to motivate useState

54

55 - HMR

- Debugging

***** REACT CONCEPTS IN TWEETER *****

CONCEPT: PROPS + CONDITIONAL RENDERING

- Data populating or displayed by UI typically comes from web API calls to server
- Could be root component fetching and data, or any other component
- Data can be passed to sub-components through PROPS
- EXAMPLE: Login vs Register (demo + code)
- Notice CONDITIONAL RENDERING on Submit button in <AuthenticationFormLayout>
- PROPS changes trigger re-render of component

CONCEPT: STATE + VIRTUAL DOM

- Review Login state variables (alias, password, rememberMe, isLoading)
- Setting any of these variables will trigger a re-render of component
- Every key stroke updates these variables, and therefore re-renders

84

85 - This seems very heavy-weight. How does
React make this frequent re-render
process efficient?

86

87 - VIRTUAL DOM

88

89

90 ***** REACT ROUTING *****

91

92 CONCEPT: What is a "route"

93

94 - route.md

95

96 CONCEPT: React Router

97

98 - Review Tweeter top-level structure and
routes: index.html, index.tsx, App.tsx

99

100 - "Route Paths" slide

101

102 - "Layout Routes" and "Layout Routes:
Outlet" slide

103

104 CONCEPT: In-app Navigation

105

106 - Demo navigation in Tweeter

107 - Navigate when switching between
login <---> register

108 - Navigate after login/register

109 - Navigate when changing between tabs

110

111 - navigate(path) in Register.tsx
(doRegister)

112

113 - <Link ...> in Register.tsx
(switchAuthenticationMethodFactory)

114
115
116
117
118
119
120
121
122
123
124

- <NavLink ...> in AppNavbar.tsx

CONCEPT: Accessing URL Location and Params

- Access URL location (path) in AppNavBar.tsx
- Access URL param
 - App.tsx to define :displayUser param,
 - useParams() in FeedScroller.tsx to access :displayUser param