Prerequisites & Case Study

AIRLINES MANAGEMENT CASE STUDY

1. Table ofContents

2. Pre-Requisites

Category / Requirements	**Skill Level / Experience**
FRONT END	
Preact/React/OJET Basics	Mandatory and implemented at least one module / feature using Preact/React/OJET
TypeScript Basics	Mandatory and implemented at least one module / feature
Routing	Mandatory and implemented static and dynamic routing
HTTP requests	Mandatory and used both GET & POST methods
Asynchronous Programming	Knowledge of asynchronous programming and libraries
Forms	Mandatory and implemented Template driven forms
Design Material	Mandatory and implemented components like Grid, List etc.
State Management	Knowledge of State management and libraries like Redux.
UI – CSS & Sass	CSS skill level must be intermediate & have knowledge of Sass
BACKEND	
RESTful API	Knowledge of RESTful APIs are required to create a backend server application

SpringBoot	SpringBoot to create the RESTful API server app
Authentication	JWT authentication or other authentication to authenticate user
CORS	Knowledge o CORS about how to stop/allow cross domain request
Response	Knowledge of proper response object
Microservices	To create the backend as collection of Microservices

3. Gained Knowledge post Case Study

Category / Requirements	**Skill Level / Experience gained**
Preact/React/OJET Intermediate	Implemented a project using Preact/React/OJET latest version and used intermediate techniques like state management, async calls, reactive programming, lazy loading
Routing	Implemented all types of routing techniques including route guards, lazy loading, redirects
HTTP requests	Handle asynchronous calls, error handling
Asynchronous Programming	Used observables, event emitters, async pipes
Forms	Implemented Reactive Forms
Design Material	Implementation experience of custom components using CDK, handle animations
State Management	Implemented store using Redux

Lazy loading	Lazy loading of modules, components, routes
UI – CSS & Sass	Implemented Sass and ensure proper structure of layout, styles and handling responsive layouts / components (small, medium, large breakpoints)
SpringBoot	How to create RESTful services using SpringBoot, create endpoints, send response as JSON/XML, map route parameters to method parameters, fetch data from request body, authentication using JWT and/or mechanisms
Microservices	How to create the backend as collection of Microservices, how Microservices talk to each other and what is API gateway etc.

4. Case Study / Requirements

4.1 Functional

Build Airline Check-In, In-Flight and ancillary management app with the following requirements

4.1.1 Airline Staff

Following features will be available for Airline staff while check-in and in-flight.

4.1.1.1 Check-In

- Select flight (from list) based on current schedule (time)
- Display flight details & seat map color coded to identify between passengers checked-in or not, passengers requiring wheel chair, passengers with infants
- Display passenger list with name, ancillary services, seat number
- Check-in passenger by select the respective seat
- Undo check-in by selecting the respective seat
- Display details of passenger like Name, Ancillary services
- Filter passengers by checked in / not, wheel chair, infant
- Change seat of passenger (through passenger list)

4.1.1.2 In-Flight

- Display flight details & seat map color coded to identify between passengers requiring special meals
- Display ancillary services requested by passenger
- Add ancillary service for a passenger
- Change meal preference for a passenger
- Add in-flight shop requests for a passenger

4.1.2 Admin

- Dashboard with option to manage passengers, ancillary services per flight
- List passengers (name, ancillary services, seat number)
- Filter passengers by missing mandatory requirements (passport, address, date ofbirth)
- Add / Update passenger – Name, passport details, address
- Add / Update / Delete ancillary services, special meals, shopping items per flight

4.2 Non-Functional

Category	**Requirement**
User Interface	Must be responsive for at least 3 breakpoints (Small, Medium, Large) Usage of Sass / Scss for styling Usage of Flex Layout (CSS)
State Management / In-Memory Store	Usage of Redux state management or OJET state management and ensure transactional & static data are managed in Memory
Accessibility	Follow W3C web standards, SEO & WCAG 2.0 Level A for accessibility **Lighthouse report >= 80 (SEO & Accessibility)**
Design Material	Usage of Material components, layouts, customized components using CDK

Performance	Usage of Asynchronous calls using Redux in handling API requests / response.
Category	**Requirement**
	Lazy Loading of modules, components & routes **Lighthouse report >= 80 (Performance)**
Best Practices	Followed best practices **Lighthouse report >=80 (Best Practices)** **Lint issues = 0**
Unit Testing	Implemented unit testing for at least one component
Forms	Made use of Forms
Authentication	Login using either of Google / Facebook / Twitter/JWT authentication
Authorization	Two roles, Admin & Airline Staff to be managed w.r.to appropriate features

5. References

Need	**Links**
User Interface (UI) Design Ideas	Color Palette, Icons, Styled components: https://coolors.co/ | https://www.huesnap.com | http://colormind.io Fonts: https://fonts.google.com Layouts: https://www.vecteezy.com
Performance	https://developers.google.com/web/fundamentals/performance/rail https://developers.google.com/web/tools/lighthouse/