Prerequisites & Case Study

201 ANGULAR PROGRAM

1. Table of Contents

2. Pre-Requisites for201

Category / Requirements	**Skill Level / Experience**
Angular Basics	Mandatory and implemented at least one module / feature using Angular 5+ version (and not lesser)
TypeScript Basics	Mandatory and implemented at least one module / feature
Angular Routing	Mandatory and implemented static and dynamic routing
Httpclient	Mandatory and used both GET & POST methods
Asynchronous Programming	Knowledge of asynchronous programming and libraries like RxJS
Forms	Mandatory and implemented Template driven forms
Angular Material	Mandatory and implemented components like Grid, List..
State Management	Knowledge of State management and libraries like Redux.
UI – CSS & Sass	CSS skill level must be intermediate & have knowledge of Sass

3. Gained Knowledge post 201

Category / Requirements	**Skill Level / Experience gained**
Angular Intermediate	Implemented a project using Angular latest version and used intermediate techniques like state management, async calls, reactive programming, lazy loading
Angular Routing	Implemented all types of routing techniques including route guards, lazy loading, redirects
Httpclient	Handle asynchronous calls, error handling

Category / Requirements	**Skill Level / Experience gained**
Asynchronous Programming	Used observables, event emitters, async pipes
Forms	Implemented Reactive Forms
Angular Material	Implementation experience of custom components using CDK, handle animations
State Management	Implemented store using ngRx/Store
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)

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 of birth)
- 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 ngRx state management and ensure transactional & static data are managed using ngRx/Store in Memory
Accessibility	Follow W3C web standards, SEO & WCAG 2.0 Level A for accessibility **Lighthouse report >= 80 (SEO & Accessibility)**
Angular Material	Usage of Material components, layouts, customized components using CDK
Performance	Usage of Asynchronous calls using ngRx in handling API requests / response.

Category	Requirement
	Lazy Loading of modules, components & routes **Lighthouse report >= 80 (Performance)**
Angular Best Practices	Followed best practices **Lighthouse report >=80 (Best Practices)** **Lint issues = 0**
Unit Testing	Implemented unit testing using Jasmine / Karma for at least one component
Forms	Made use of Angular Reactive Forms
Authentication	Login using either of Google / Facebook / Twitter
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/
Tutorials (Pluralsight)	https://app.pluralsight.com/library/courses/angular-architecture-best-practices/

	https://app.pluralsight.com/library/courses/rxjs-getting-started https://app.pluralsight.com/library/courses/angular-ngrx-getting-started https://app.pluralsight.com/library/courses/angular-routing/ https://app.pluralsight.com/library/courses/angular-services https://app.pluralsight.com/library/courses/angular-2-reactive-forms https://app.pluralsight.com/library/courses/angular-cli https://app.pluralsight.com/library/courses/unit-testing-angular https://www.pluralsight.com/courses/modern-web-layout-flexbox-css-grid
Tools & Libraries	Angular (latest) : https://angular.io/guide/quickstart ngRx (State management) : https://ngrx.io Angular Material : https://material.angular.io Authentication (Social) : https://github.com/abacritt/angularx-social-login#readme