Program on Building Modern Web Apps with NextJS – (Express Mode)

# Foundation Program - Optional

*Note:* Basis the Calibration Result. Learners may do this if they need a refresher program on Building React Apps.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Layer** | **Week** | **Module** |  | **Indicative Coverage** |
| **Foundation:** Building  Single Page Application with ReactJS | 1.5 | Building  Responsive and  Interactive  Web Pages  Using HTML5, CSS3,  BootStrap and JavaScript | • •  •  •  •  •  • | Setup Environment, Working with Git & Gitlab, Setup IDE Building static and responsive Web pages using HTML5 / CSS3, FlexBox, Semantic Tags, Hero Unit, Grid based layouts.  Building Responsive Web Pages using BootStrap (media queries, viewport) Introduction to JavaScript (Language basics, functions, objects, arrays) &  Modern JavaScript, ES6 (Closure, Async, Callback), Promises, DOM,  Java Script, Prototypes in JS  Build Interactive & Responsive Web Page |
| Working with  Server- Side  JavaScript  (Node.js) | •  • | Features of Node.js, Nodejs Architecture - callback concepts, blocking code and non-blocking code, inbuilt core modules, export statement, creating and including custom modules  HTTP module, Node.js Web server, Read & Split query string, working with file system, creating Buffers and Streams |
|  |  |  | • | NPM & Packages, Downloading package, using a package.json, node\_modules |
|  | 1.5 | Building Single Page  Applications  Using React | •  •  •  •  • | Building Interactive Web pages with React JS; States and Props; Webpack; Component Life Cycle.  Thinking in React, React Routers. ES6, JSX – Competent;  Rendering Elements, Components and Props, State and Lifecycle. Handling Events, Conditional Rendering, Lists and Keys, Forms Lifting State Up, Composition vs Inheritance.  Test Automation using JEST and Enzyme  React Routers, Rendering Elements, Components and Props,  Functional Components, State and Lifecycle |
|  |  |  | • | Handling Events, Conditional Rendering, Lists and Keys, Forms Lifting State Up, React-hooks & Context, Unit Testing ReactJS Apps using |
|  |  |  | • | Enzyme / JEST, Integrating APIs with Fetch/axios – GET, POST, PUT,DELETE, Pure Components |
|  |  |  | • | Redux Data, Async and Data Fetching |
|  |  |  | • | RxJS & Redux-Observables, Reducers & Actions, Implement Undo History, ImmutableJS, Performance and Normalizing Data,  Asynchronous Operations, Memoization Redux-Thunk; Redux-Saga |
|  |  |  | • • | **Project Stage 1: Build a Single Page Application** |

# Core Program – Build and Deploy Modern Day Applications with NextJS (With Cloud)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Layer** | **Week** | **Module** |  | **Indicative Coverage** |
| **Core:**  Build and  Deploy  Modern day  NestJS &  NextJS  Applications | 1 | NestJS | * Introduction to NestJS and Installing the NestJS CLI (command-line interface) * NestJS Application, Working with NestJS CLI API Design Platform / API Client - Install and Setup Insomnia * Running NestJS in Development Mode * Creating a Basic Controller * Use Route Parameters * Handling Request Body / Payload * Response Status Codes * Handling Update and Delete Requests * Implement Pagination with Query Parameters * Creating a Basic Service * Send User-Friendly Error Messages * Encompass Business-Domain in Modules * Introduction to Data Transfer Objects * Validate Input Data with Data Transfer Objects * Handling Malicious Request Data * Auto-transform Payloads to DTO instances | |
| 1 | NextJS | * Introduction to Next JS * Creating Next App • File based Routing. * Next Components (Server/Client) * Page Layouts and Head Meta * Fetching data in Server Component * Handling Loading and error State * Implementing Authentication * Global state with Context API | |
| Cloud  Fundaments | 0.5 | Cloud Fundamentals | * Basics Cloud (AWS) Overview * Virtualization Basics, RDB Cloud Fundamentals (SaaS, PaaS, IaaS), Best Practices with Git and Engineering Discipline on Cloud workflow, Private   / Public / Hybrid Clouds, Storing (S3 Buckets, RDS, EC2, VPC, EBS, EKS), AWS CloudFront | |
| Capstone  Project | 0.5 | Apply &  Demonstrate competency gained | • **Project Stage 2** – Develop, test and deploy a complete web application. | |