

©2020 Impetus Technologies, Inc.

You are prohibited from making a copy or modification of, or from redistributing, rebroadcasting, or re-encoding of this content without the prior written consent of Impetus.

This presentation may include images from other products and services. These images are used for illustrative purposes only. Unless explicitly stated there is no implied endorsement or sponsorship of these products by Impetus. All copyrights and trademarks are property of their respective owners.

# UIUX - RTB C12 - React

Manual & Case Study

# Skill Manual

# Objective

- Readiness of an individual to be deployed on a project for React application development work
- Be able to share knowledge/updates with the group/organization around React
- Ability to share estimates for application/products based on React

# Topics To Cover

## Milestone - 1

### 1. React QuickStart & JSX

- React Architecture, Fundamentals and ReactDOM
- Setting React Environment and basic configuration to start with React
- Why JSX?
- JSX features and usage
- Embedding Expressions in JSX and rendering elements

### 2. Component, Forms, Events

- Understanding React components and structure
- Class based Components and rendering
- States, Props, Props validation and setState
- Composing Components and Conditional Rendering
- Controlled Components, Handling Multiple Inputs and Uncontrolled components
- ForceUpdate, findDOMNode, React Memo, React lazy and suspense

#### **State & Lifecycle - *Class Component***

- The component Lifecycle methods (Mounting, Updating, Unmounting)

**Functional Components** – Structuring, hooks ( useState, useEffect, useSelector, useDispatch etc..)

#### **Events**

- What is event handler?
- Passing Arguments to Event Handlers
- Child events, Refs, Keys

# Topics To Cover

## Milestone - 2

### 1. Router

- Declarative routing and Route configurations
- BrowserRouter vs HashRouter vs MemoryRouter, Link vs NavLink
- Redirect, Router and StaticRouter
- Switch and Hook:- useHistory, useLocation
- Nesting of Routes, IndexRoutes with React Router
- Fetching Data with Lifecycle Methods
- Forms and Form Submission
- Dynamic Route Params
- Loading Data on Render
- Handling Null Props

### 2. Next Generation JavaScript & Advance React Fundamentals

- Arrow Function
- Export and Imports
- Spread and Rest Operator
- Best practices with ES and feature of latest version of ES over old versions
- Higher Order Components
- Context API
- Fragment
- Pure Component
- Error Boundaries
- Features introduced in new/latest version of React over older versions

# Topics To Cover

## Milestone - 3

### 1. Redux

- What is Redux and architecture?
- Reducer, Actions, Store, Data flow and Unidirectional dataflow
- Reducers and Containers
- Actions (and ActionCreators) and Dispatching Actions
- Filtering Redux Data and Sorting Redux Data
- The single immutable state tree
- State changes with actions
- Pure, Impure and Reducer functions
- Container vs Presentational components
- Passing Control to Redux Form and Catching Data with Posts Reducer

### 2. Redux Middleware & Creating APIs

- Introduction to Middleware
- Middleware: Thunk vs saga.
- Working with API's
- Ajax Requests with Axios
- Redux-Promise in Practice
- Redux-Promise Continued
- Forwarding Actions
- State Validation Middleware
- Async Action, Async Flow

# Topics To Cover

## Milestone - 4

### Unit Testing

- Setup of unit testing framework – Jest, Enzyme etc.
- Mounting
- Shallow
- Render
- Snapshot creation
- Testing the states
- Mock Function
- Testing Class Names
- Simulating Events
- Expecting Child Elements
- Testing Controlled Components
- Form Submit Event



# Topics To Cover

## Milestone - 5

### Packaging & Deployment

- Webpack integration – bundling, minification, dev and prod env setup
- UTC coverage report configuration and generation
- Code Linting configuration and report generation
- Configuration using loaders
- HMR / React Hot Loader plugin

### Material UI for React (Good to have)

- Material-UI setup in project
- Styles
- Layout
- Modals
- Form components
- Buttons
- Cards
- Chips
- Grid list integration

# Expectations

Team Members post the intervention should be able to conduct knowledge sharing session and complete the Assessments. We will be sharing the mode of assessments in the group/team level meeting.

# Case Study

# Problem Statement / Use Case

Details:

User can have two types of role: Admin/Employees

## 1) Role: Admin

1. User should sign up as Admin
2. Employee Portal
  - (a) Admin can add an Employee in Canteen management system.
  - (b) Admin can view Employee List with their Employee Ids registered in Canteen Management System.

- (c) Admin can search any Employee, also can filter the List with Id, Name etc.
- (d) Admin can add money to the canteen account of any Employee.
- (e) Admin can check balance of the Employee account.

### 3. Item Portal

- (a) There should be two lists available, one – Items Of The Day

Second- Master List of Total Items that can be available in Canteen

- (b) Items Of The Day would be reset daily and Items available in Canteen for the day corresponding to Quantity would be added/updated.
- (c) Admin can view list Of All Items In Canteen corresponding to their price.
- (d) When adding Item of the Day, Admin would be given option to select out of Total Items that can be available In Canteen, price would be auto populated then.
- (e) Only quantity is an editable attribute for any Items of the Day.
- (f) Admin can add/update/delete items with Item name, Quantity, Price etc.

4. Admin can deduct money corresponding to any employee if employee is purchasing anything through Admin.

Quantity of that particular item should be updated in portal. Like the quantity was 10 initially and purchased by two users, should be 8 now.

Any article could only be purchased if its quantity  $> 0$ .

5. Admin can view Credit Details Of Company , Transaction details, Items available.

## 2) Role: Employee

1. User can Sign Up In the account with Employee Id, Name etc., (once signed up user would be available in Employee List of Admin)
2. User can check balance in the Canteen account.
3. User can view Items of The Day Available (would not able to view any item whose quantity is 0).
4. User can add money to the his/her canteen account.
5. User can view the passbook (details of all transactions on his account) and source of deduction (Admin/Employee).
6. User can also view menu details (corresponding to price ) for the day present in canteen.
7. User can buy item from Menu List.



### 3) Good To Have Functionalities:

- a) On buying the item by employee , amount would be deducted from employee's account and item would be added in transaction details and quantity for that particular item would be added (Quantity would be reflected only for Admin).
- b) Admin can only delete or update Item from Master List Of Items, if that item is not available in Items Of The Day.

## **Generic Points for whole App:**

- 1) Pagination should be included in all lists created.
- 2) Search should be enabled for all lists.
- 3) All tables should be sortable.
- 4) At least 1 service of the application should also be available as web services (REST)
- 5) Web storage for Offline data.

# Expected Deliverables

1. Application Demo covering all functionalities
2. Application codebase

Below are the sample view that can be shown as dashboard for both Admin and Employee.  
Please refer below.

DASHBOARD VIEW FOR ADMIN:

View should be interactive and clickable enough to get you to the detailed version of entity.

<div><div>Users</div><div>1) ABC</div><div>2) DEF</div><div>Add User</div></div>	<div><div>Items Of The Day</div><div>1) XYZ</div><div>2) UVW</div><div>Add Item Of The Day</div></div>
<div><div>Transaction Details</div><div>1) Rs. 100 - XYZ - Date/Time</div><div>2) Rs. 100 - UVW - Date/Time</div></div>	<div><div>Items Available</div><div>1) 123</div><div>2) 456</div><div>Add Item</div></div>

DASHBOARD VIEW FOR EMPLOYEE:

View should be interactive and clickable enough to get you to the detailed version of entity.

<div><div>Items Of The Day</div><div><div>1) XYZ</div><div>—</div><div>Rs. 100</div><div>&lt;Further details like Image can be shown here&gt;</div></div><div><div>2) UVW</div><div>—</div><div>Rs. 100</div></div><div>Buy Item</div></div>			
<div><div>My Transactions</div><div><div>1) Rs. 100</div><div>-</div><div>XYZ</div><div>-</div><div>Date/Time</div></div><div><div>2) Rs. 100</div><div>-</div><div>UVW</div><div>-</div><div>Date/Time</div></div></div>		<div><div>Account Passbook</div><div><div>1) Rs. 100</div><div>Data/Time</div><div>Credit</div></div><div><div>2) Rs. 200</div><div>Date/Time</div><div>Debit</div></div><div>Add Money</div></div>	

# Assumptions

None

# Quality Expectations (Do's / Don'ts)

Best practices need to be followed, for more the details:-

<https://reactjs.org/docs/getting-started.html>

**Thanks.**