|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Cognizant Academy**  **eCommerce App**  **ReactJS Specification**  **Version 0.1**   |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** |  |  |  | | **Role** |  |  |  | | **Signature** |  |  |  | | **Date** |  |  |  | |
|  |
|  |

Table of Contents

[1.0 Introduction 3](#_Toc61006270)

[1.0 Purpose of this document 3](#_Toc61006271)

[2.0 Definitions & Acronyms 3](#_Toc61006272)

[3.0 Project Overview 3](#_Toc61006273)

[4.0 Scope 3](#_Toc61006274)

[5.0 Intended Audience 3](#_Toc61006275)

[6.0 Hardware and Software Requirment 4](#_Toc61006276)

[2.0 Application UI Layout 4](#_Toc61006277)

[3.0 Architecture 5](#_Toc61006278)

[4.0 Project structure 5](#_Toc61006279)

[5.0 MenuComponent 5](#_Toc61006280)

[6.0 S001 – ProductListComponent 6](#_Toc61006281)

[7.0 S002 – CartComponent 7](#_Toc61006282)

[8.0 S003 – BillComponent 8](#_Toc61006283)

9[.0 Submission 12](#_Toc61006287)

[10.0 Change Log 12](#_Toc61006288)

# Introduction

## Purpose of this document

The purpose of this document is outlining the specification for a ReactJS application which allows users to view products available, add items to the cart,view cart items and display bill.

## Definitions & Acronyms

|  |  |
| --- | --- |
| **Definition/ Acronym** | **Description** |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheet |
| RWD | Responsive Web Design |
| UX Design | User Experience Design |

## Project Overview

Developing ecommerce application UI using ReactJS based on the user stories provided in the list below



## Scope

Developing an application front-end using ReactJS library

## Intended Audience

* Product Owner
* Scrum Master
* Application Architect
* Project Manager
* Test Manager
* Development Team
* Testing Team

## Hardware and Software Requirement

1. Hardware Requirement:
   1. Developer PC with 8GB RAM
2. Software Requirement
   1. NodeJS
   2. Visual Studio Code
   3. Internet explorer/Google Chrome

# Application UI Layout

App Component

Menu Component

Placeholder for displaying other components

Figure 1: Application UI Layout

# Architecture

Database

REST API

React Front-End

Figure 2: Architecture Diagram

# Project structure

1. Create a new ReactJS application
2. Create a sub-folder with-in src named as **“components”**
3. Each component must be created in a separate **<componentname.js>** file with-in **“component”** sub-folder.
4. Create a sub-folder within src names as “**services**”
5. Create a service called **EcommerceService.js** through which the components will interact with each other.

# MenuComponent

1. Create MenuComponent inside **“menucomponent.js”**, in the components sub-folder.
2. MenuComponent shouldn’t have any state and must be created as a functional component insteadd of class-based component.
3. MenuComponent will be used for display a list of selection to the user based.
4. On clicking each menu item an appropriate react component must be displayed below the menu component inside the placeholder
5. For an user menu component will display only the following menu items
   * 1. Products
     2. Cart
6. Along side the menu items menu component will also display the application name

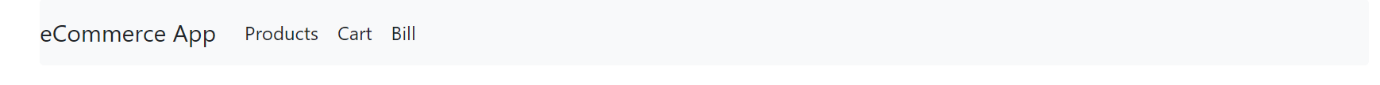


Figure 3: Menu Component

# S001 – ProductListComponent

1. Create **ProductListComponent** with in **components** folder inside the **“productlistcomponent.js”** file
2. ProductListComponent should be a class component. It can receive the required data through the props and maintain its own state.

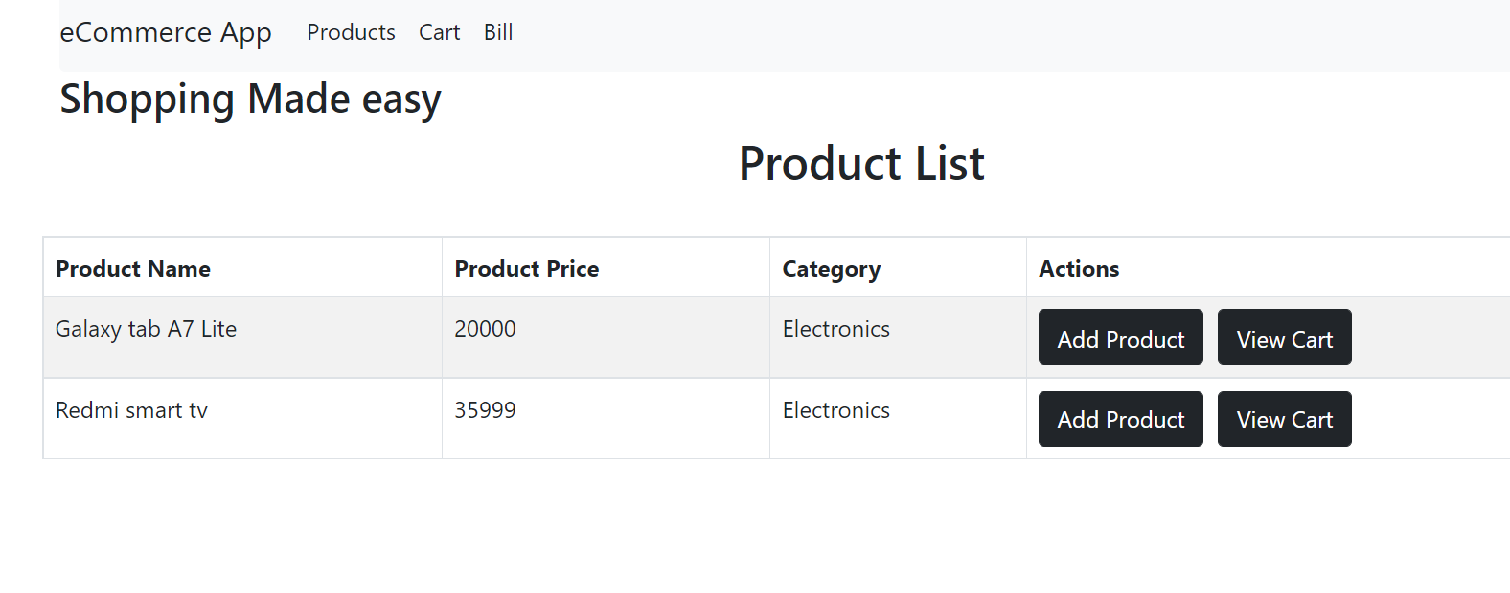


Figure 5: ProductListComponent

1. ProductListComponent will interact with the RESTAPI and fetch the product data and render it as a list .
2. Ensure that each item in the list has a unique key and list is created using the map method.

# S002 – CartComponent

1. Create a new component inside **components** folder with the name as CartComponent in a file named cartcomponent**.js.**
2. CartComponent will interact with the REST API endpoint to get the list of products ies added to cart of the user.

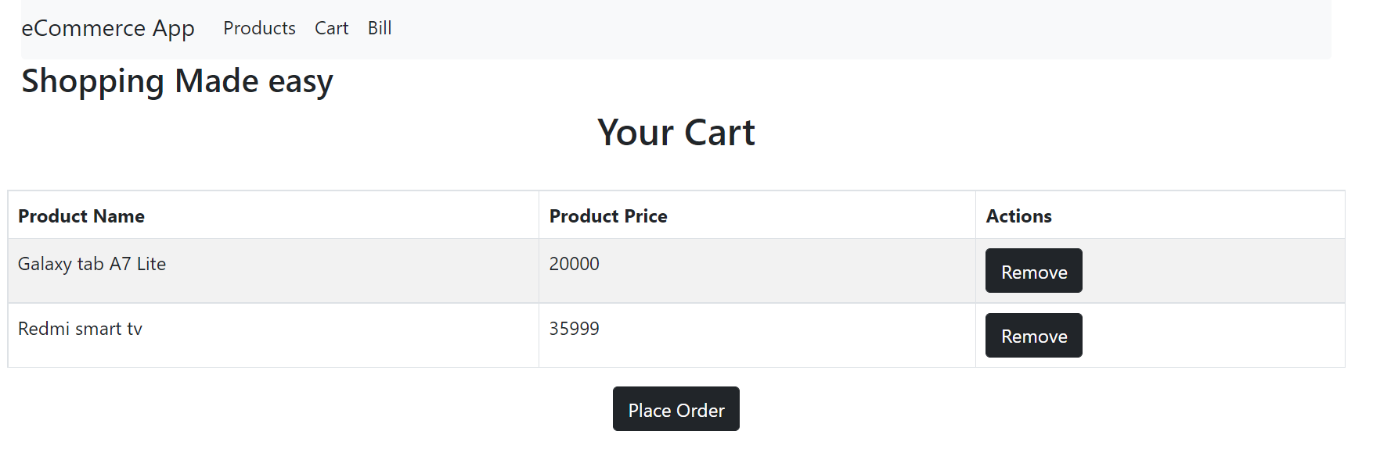


Figure 6: CartComponent

1. CartComponent will have a “Remove” button which will allow the current user to remove the items from the cart list.
2. On clicking the Remove button inside CartComponent, a REST API call must be made from CartComponent.

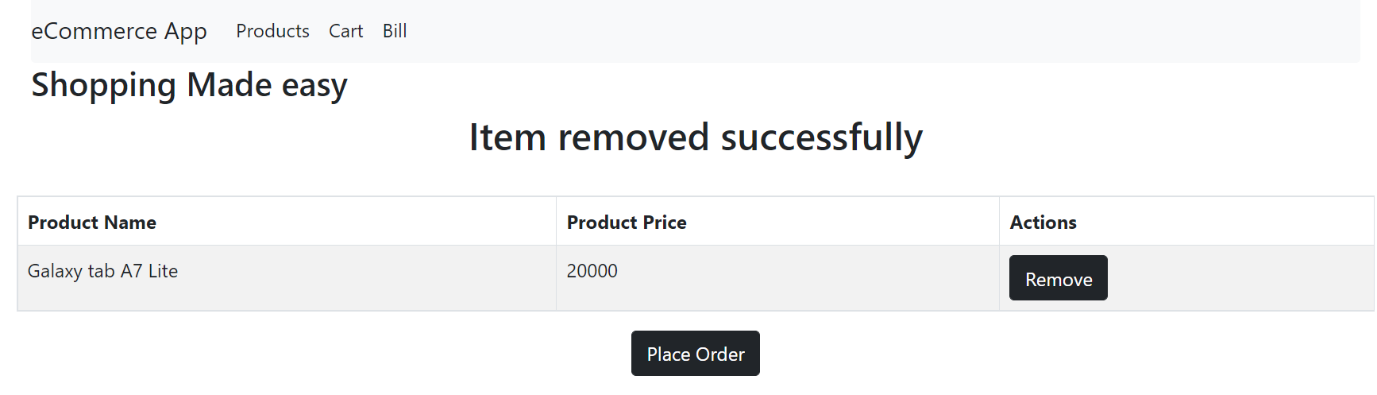


Figure 7: CartComponent

1. After successful removal of the product from cart message must be displayed to the user.
2. If there are no products added to the cart list display the message as “No items added to the cart”

Figure 7: CartComponent

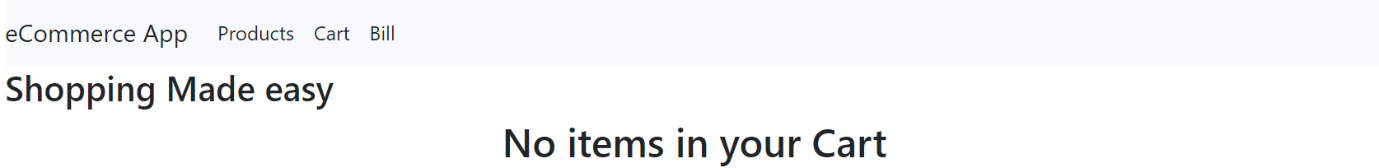
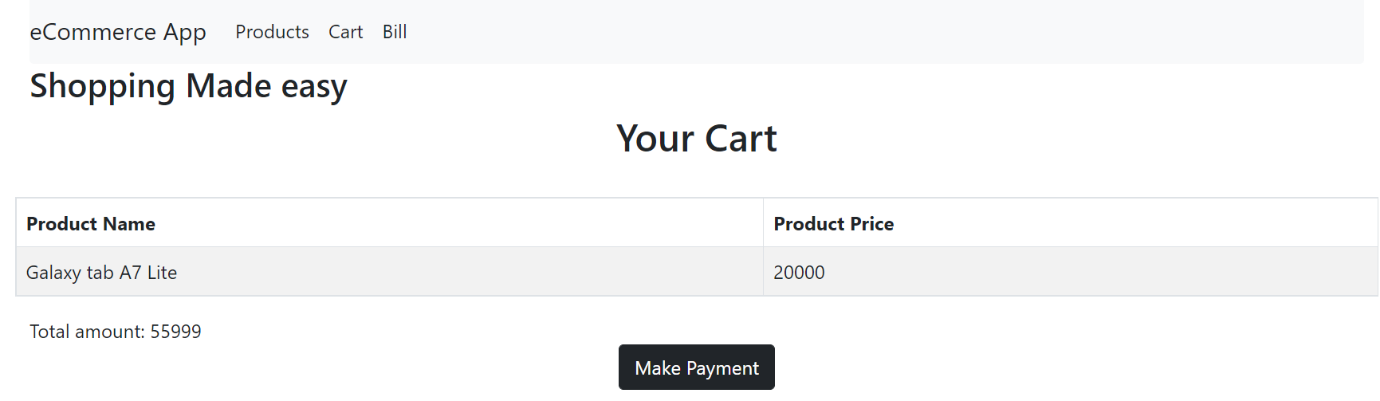


Figure 7: CartComponent

# S003 – BillComponent

1. Go to **components** folder and create a new component named **BillComponent** inside the file **BillComponent.js**
2. This component will render the bill for the user.
3. After order is placed, user can make payment.



# Submission

1. Use git add, commit and push commands to upload your code into remote GltLab repository.
2. During commit, give the commit message as “ReactJS ecommerce”.

# Change Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Changes Made** | | | |
| V1.0.0 | Initial baseline created on <dd-Mon-yy> by <Name of Author> | | | |
| Vx.y.z | <Please refer the configuration control tool / change item status form if the details of changes are maintained separately. If not, the template given below needs to be followed> | | | |
| **Section No.** | **Changed By** | **Effective Date** | **Changes Effected** |
|  |  |  |  |