**PART A**

**Experiment No. 05**

**A.1 Aim:** Create react application to implement components and use props

**Objective:** To learn and understand how to pass prop and create reusable components that structure your app.

**A.2 Prerequisite:**

HTML, CSS, Javascript

**A.3 Outcome:**

Upon completion of this section, you will be able to:Understand and implement react library as a front end

**A.4 Theory**

React.js is a popular JavaScript library for building user interfaces, particularly single-page applications where you need a fast, interactive user experience. It's developed by Facebook and allows developers to create large web applications that can change data, without reloading the page. The main goal is to be fast, scalable, and simple. It works only on user interfaces in the application; this corresponds to the view in the MVC template.

Components: React is all about components. You build your UIs out of components. Components are reusable and can be nested within each other. A React application is typically a tree of components. They are of 2 types: functional and class

JSX: JSX is a syntax extension for JavaScript. It looks like HTML but works inside JavaScript. React uses JSX to describe UIs. JSX is optional and not required to use React, but it's recommended as it visually resembles the output it generates.

Props: Short for properties, props are how you pass data from a parent component to a child component. Props are read-only and should not be modified by the child component.Tasks to be completed

Example

import React from 'react';

import ReactDOM from 'react-dom';

import Greeting from './Greeting'; // Assuming Greeting is saved in a file named Greeting.js

function App() {

return (

<div>

<Greeting name="Alice" />

<Greeting name="Bob" />

</div>

);

}

ReactDOM.render(<App />, document.getElementById('root'));

// Array of favorite books

const books = [ { id: 1, title: '1984', author: 'George Orwell' }, { id: 2, title: 'To Kill a Mockingbird', author: 'Harper Lee' }, { id: 3, title: 'The Great Gatsby', author: 'F. Scott Fitzgerald' } ];

{books.map((book) => ( <BookItem key={book.id} title={book.title} author={book.author} /> ))}

**Problem**: Create a React.js Application for Displaying a Favorite Book List

**Create a react js favourite book list application to demonstrate component creation and rendering with props. (Use CSS wherever required and** display book images and prices**)**

Requirements:

1. Components:
   * Create three components:
     + App: The main component that holds the list of books.
     + BookList: A component that receives the list of books as props and maps through them.
     + BookItem: A component that displays individual book details including title, author, image, and price.
2. Props:
   * Use props to pass information about each book (title, author, image, and price) from App to BookList and finally to BookItem.
3. Images:
   * Each book should have a cover image displayed. The images should be stored locally in the public/images folder and referenced within the components.
4. CSS Styling:
   * Style the application using CSS to create a visually appealing layout.
   * Ensure the layout is responsive, with books displayed in a grid or column structure depending on the screen size.
5. Book Details:
   * Each book in the list should display the following:
     + Title of the book.
     + Author of the book.
     + Cover Image of the book.
     + Price of the book.

**PART B**

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

|  |  |
| --- | --- |
| Roll No. : S020 | Name: Husain Chhil |
| Class : MBA.Tech DS Sem V | Batch : J1 |
| Date of Experiment : 10.03.24 | Date/Time of Submission : 10.03.24 |
| Grade : |  |

**B.1 Code:**

Book.js

import "../App.css";

function Book({ *title*, *author*, *price*, *image\_path* }) {

  return (

    <div *class*="book">

      <img *src*={*image\_path*} *alt*={*title*} />

      <h2>{*title*}</h2>

      <h3>{*author*}</h3>

      <h4>{*price*}</h4>

    </div>

  );

}

export default Book;

BooksList.js

import Book from "./Book";

import "../App.css";

function BookList({ *books* }) {

  return (

    <div *class*="bookslist">

      {*books*.map((*book*) => (

        <*Book*

*title*={*book*.title}

*author*={*book*.author}

*price*={*book*.price}

*image\_path*={*book*.image\_path}

        />

      ))}

    </div>

  );

}

export default BookList;

App.js

import BookList from "./components/BookList";

import "./App.css";

function App() {

  const books = [

    {

      title: "The Alchemist",

      author: "Paulo Coelho",

      price: "$10",

      image\_path:

        "https://m.media-amazon.com/images/I/61HAE8zahLL.\_AC\_UF1000,1000\_QL80\_.jpg",

    },

    {

      title: "The Little Prince",

      author: "Antoine de Saint-Exupéry",

      price: "$15",

      image\_path:

        "https://m.media-amazon.com/images/I/61NGp-UxolL.\_AC\_UF1000,1000\_QL80\_.jpg",

    },

    {

      title: "The Book Thief",

      author: "Markus Zusak",

      price: "$12",

      image\_path:"https://m.media-amazon.com/images/I/91JGwQlnu7L.\_AC\_UF1000,1000\_QL80\_.jpg",

    },

  ];

  return (

    <div *class*="App">

      <h1 *class*="header">Bookstore</h1>

      <*BookList* *books*={books} />

    </div>

  );

}

export default App;

App.css

*.bookslist* {

  display: flex;

  flex-wrap: wrap;

  color: #333;

}

*.header* {

  background-color: #333;

  color: #fff;

  padding: 10px;

  text-align: center;

  border-radius: 5px;

}

*.book* {

  width: 30%;

  text-align: center;

  margin: 10px;

  padding: 10px;

  border: 1px solid #ccc;

  border-radius: 5px;

  box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);

}

*.book* img {

  width: 50%;

  height: auto;

}

*.App* {

  font-family: "Trebuchet MS", "Lucida Sans Unicode", "Lucida Grande",

    "Lucida Sans", Arial, sans-serif;

}

**B.2 Output**

