

Learner Assignment Submission Format

Learner Details

- **Name:**H O Akash
 - **Enrollment Number:**SU625MR015
 - **Batch / Class:**MERN Stack
 - **Assignment:** React JS
 - **Date of Submission:** 30/07/2025
-

Problem Solving Activity 1.1

Product Display Card

Build a ProductCard component for an e-commerce platform.

- Props: title, price, imageUrl, inStock
- Display the product image and title
- Show the price formatted as \$xx.xx
- If inStock is false, show a red “Out of Stock” label

2. Algorithm

1. App component:

1. Initialize the list of Card components.
2. Display the list of Card components.

2. Card component:

1. Initialize the props (name, price, availability, imageurl).
2. Initialize the clicked state to false.
3. When the "Add To Cart" button is clicked, set the clicked state to true.
4. Check the availability of the product.

5. If the availability is "Available:0", display the "Out of Stock" message.
6. If the availability is not "Available:0", display the "Item Added Successfully" message.

3. Pseudocode

BEGIN App component

 INITIALIZE list of Card components

 DISPLAY list of Card components

END

BEGIN Card component

 INITIALIZE props (name, price, availability, imageurl)

 INITIALIZE clicked state to false

 WHEN "Add To Cart" button is clicked

 SET clicked state to true

 END WHEN

 IF availability is "Available:0"

 DISPLAY "Out of Stock" message

 ELSE

 DISPLAY "Item Added Successfully" message

 END IF

END

4. Program Code (App component)

```
// import { useState } from 'react'

// import reactLogo from './assets/react.svg'

// import viteLogo from '/vite.svg'

import './App.css'

import Home from './Home'

import './Card.css'

import Card from './Card'


function App() {

  return (

    <div className='a'>

      <Card name="Laptop" price="Price:$1500" availability="Available:22"
imageurl={"laptop.jpg"}/>

      <Card name="Mobile" price="Price:$700" availability="Available:33"
imageurl={"mobail.jpg"}/>

      <Card name="Charger" price="Price:$50" availability="Available:0"
imageurl={"charger.jpg"}/>

      <Card name="Bag" price="Price:$200" availability="Available:11"
imageurl={"bag.jpg"}/>

      <Card name="Pant" price="Price:$50" availability="Available:22"
imageurl={"pant.jpg"}/>

      <Card name="Shirt" price="Price:$30" availability="Available:33"
imageurl={"shirt.jpg"}/>

      <Card name="Bottel" price="Price:$5" availability="Available:0"
imageurl={"bottel.jpg"}/>

    </div>

  )
}
```

```

<Card name="Headphone" price="$Price:200" availability="Available:11"
imageurl={"headphone.jpg"}/>

</div>

)

}

export default App

```

5. Program Code (Card component)

```

import React, { useState } from 'react';

const Card = ({name,price,availability,imageurl}) => {

  const [clicked, setClicked] = useState(false);

  const handleClick = () => {

    setClicked(true);

  };

  return (

    <div className='Container'>

      <div className='inner'>

        <p><img src={imageurl} alt="ph" height={"300px"}
width={"250px"}/></p>

```

```

    <h3>{name}</h3>

    <p>{price}</p>

    <p>{availability}</p>

    {clicked && availability === "Available:0" && (
        <p className="out-of-stock" style={{color:'red'}}>Out of Stock</p>
    )}

    {clicked && availability !== "Available:0" && (
        <p className="in-stock" style={{color:'green'}}>Item Added
        Succesfully</p>
    )}

    { /* {!clicked && (
        <p className="stock-status">check</p>
    )} */}

    <button type="submit" onClick={handleClick}>Add To Cart</button>

</div>

</div>

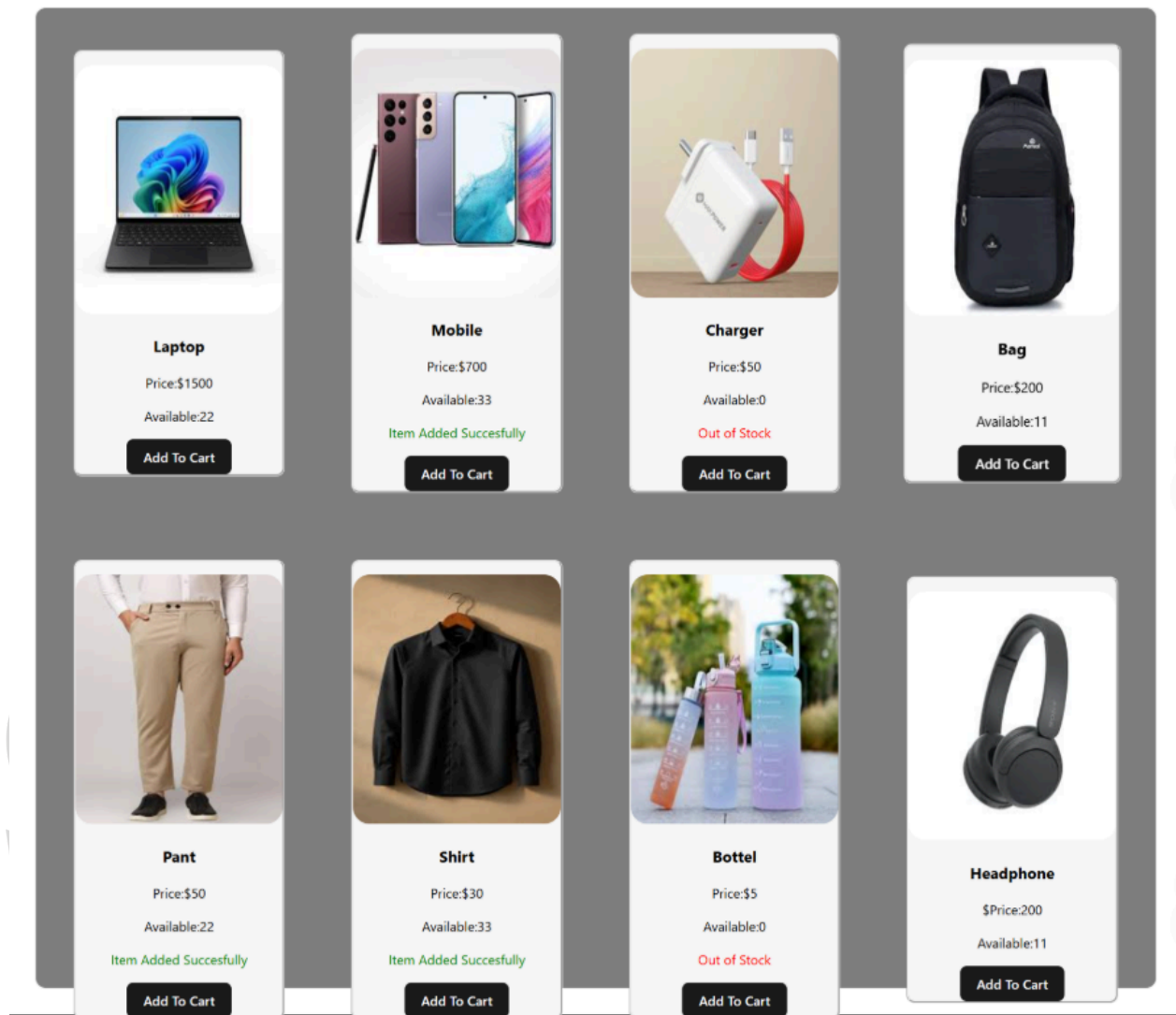
);

};

export default Card

```

6. Screenshots of Output



7. Observation / Reflection

- The given code is for a simple e-commerce application built using React. The application displays a list of products, each represented by a Card component. When a user clicks the "Add To Cart" button on a Card, it checks the availability of the product and displays a message accordingly.
- Overall, the code is a good starting point, but there are a few areas that could be improved to make it more robust and reliable.

