**Practical No: 05**

**Aim:  Event Handling**

**i)  Define a form submission event handler is using an arrow function that prevents the default form submission behavior using \*\*event.preventDefault()\*\*. Update the input value using the \*\*handleInputChange\*\* function, which is called on every input change event.**

import React, { useState } from 'react';

function P5i() {

const [inputValue, setInputValue] = useState('');

const handleSubmit = (event) => {

event.preventDefault();

console.log('Submitted value:', inputValue);

};

const handleInputChange = (event) => {

setInputValue(event.target.value);

};

return (

<form onSubmit={handleSubmit}>

<input type="text" value={inputValue} onChange={handleInputChange} />

<button type="submit">Submit</button>

<h2>submitted value {inputValue}</h2>

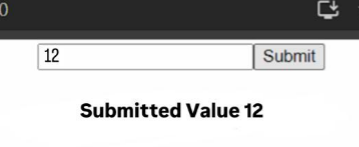
</form>

);

}

export default P5i

**Output:**



**ii) Define a keydown event handler is using an arrow function that checks if the Enter key is pressed and logs the input value to the console. Update the the input value using the \*\*handleInputChange\*\*function, which is called on every input change event.**

import React, { useState } from 'react';

function KeyDown() {

const [inputValue, setInputValue] = useState('');

const handleKeyDown = (event) => {

if (event.key === 'Enter') {

console.log('Submitted value:', inputValue);

}

};

const handleInputChange = (event) => {

setInputValue(event.target.value);

};

return (

<div>

<input type="text" value={inputValue} onKeyDown={handleKeyDown} onChange={handleInputChange} />

<p>{inputValue}</p>

</div>

);

}

export default KeyDown

**Output:**

****

**iii. Take an input field to add an element into array using states.**

import React from 'react'

import { useState } from 'react';

function MultipleValuesArr() {

const friendsArray = ["Star", “Moon”,"Juipter", "Twinkle"]

const [friends, setFriends] = useState(friendsArray);

const handleFunction = () => {

const newfriend = document.getElementById("friend").value

setFriends([...friends, newfriend])

}

return (

<div>

<h1>List</h1>

<ul>

{friends.map( (f, index)=>

<li key = {index}>

{f}

</li>

)}

</ul>

<input type = "text" id ="friend" placeholder='Enter friend name'/>

<button onClick={handleFunction}>Add Values</button>

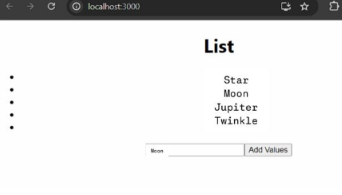
</div>

)

}

export default MultipleValuesArr

**Output:**



**iv. Take a button to add users, remove user and add user after a specific element.**

import React, { useState } from "react";

function AddRemoveArr() {

const [users, setUsers] = useState([

{ id: 1, username: "User 1" },

{ id: 2, username: "User 2" }

]);

const addUser = () => {

const newUser = {

id: ` ${users.length+ 1}`,

username: `User ${users.length + 1}`

};

const newUsers = [...ule.log(newUsers);

sers, newUser];

conso

setUsers(newUsers);

};

const removeUser = (index) => () => {

const newUsers = [...users];

console.log(newUsers);

newUsers.splice(index, 1);

setUsers(newUsers);

};

const addAfter = (index) => () => {

const newUser = {

id: `${users.length + 1}`,

username: `User ${users.length + 1}`

};

const newUsers = [...users];

newUsers.splice(index + 1, 0, newUser)

console.log(newUsers)

setUsers(newUsers);

};

return (

<>

<button onClick={addUser}>Add User</button>

{users.map((user, index) => (

<div className="user" key={user.id}>

{user.username}

<button onClick={removeUser(index)}>Remove User</button>

<button onClick={addAfter(index)}>Add immediate next User</button>

</div>

))}

</>

);

}

export default AddRemoveArr

**Output:**