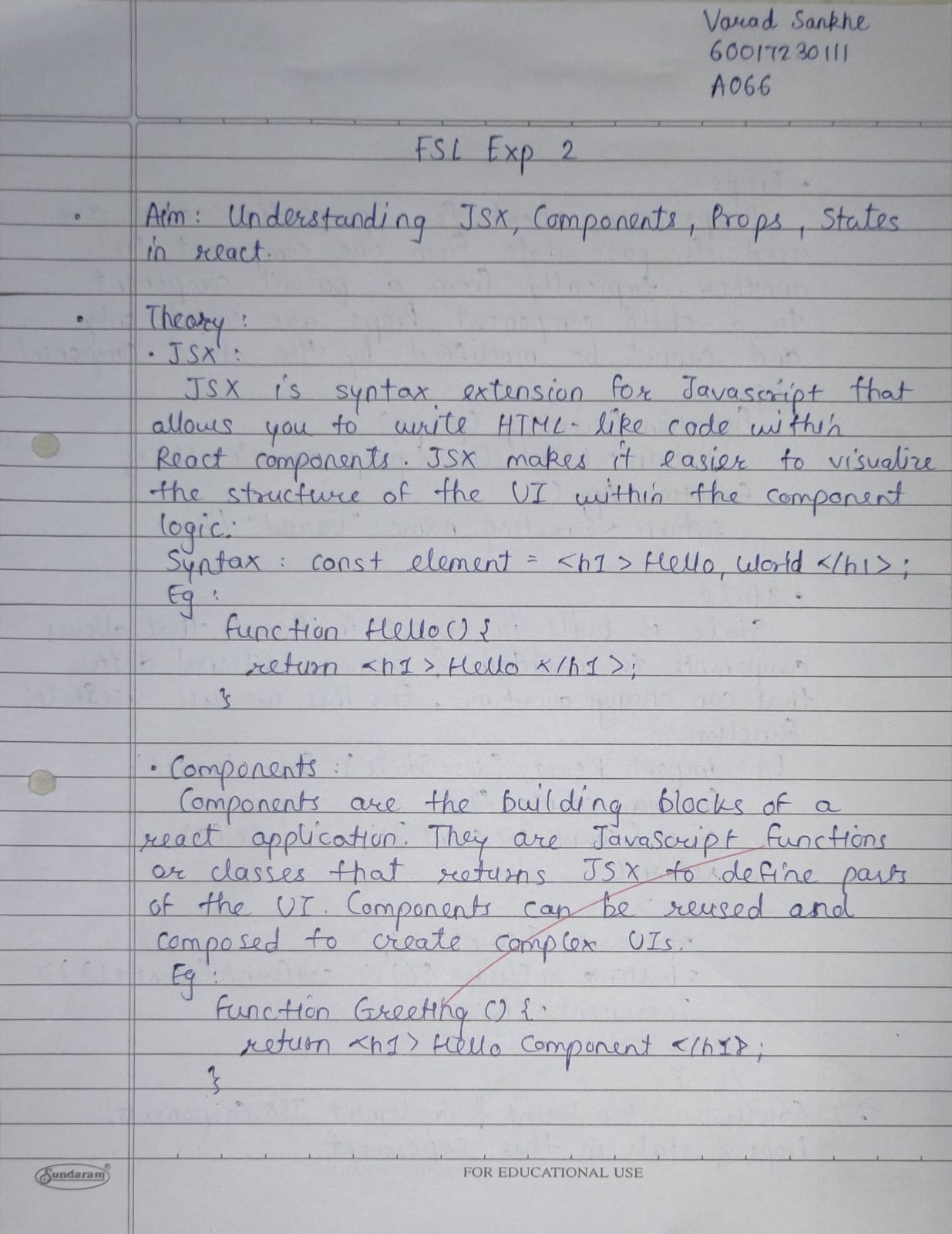
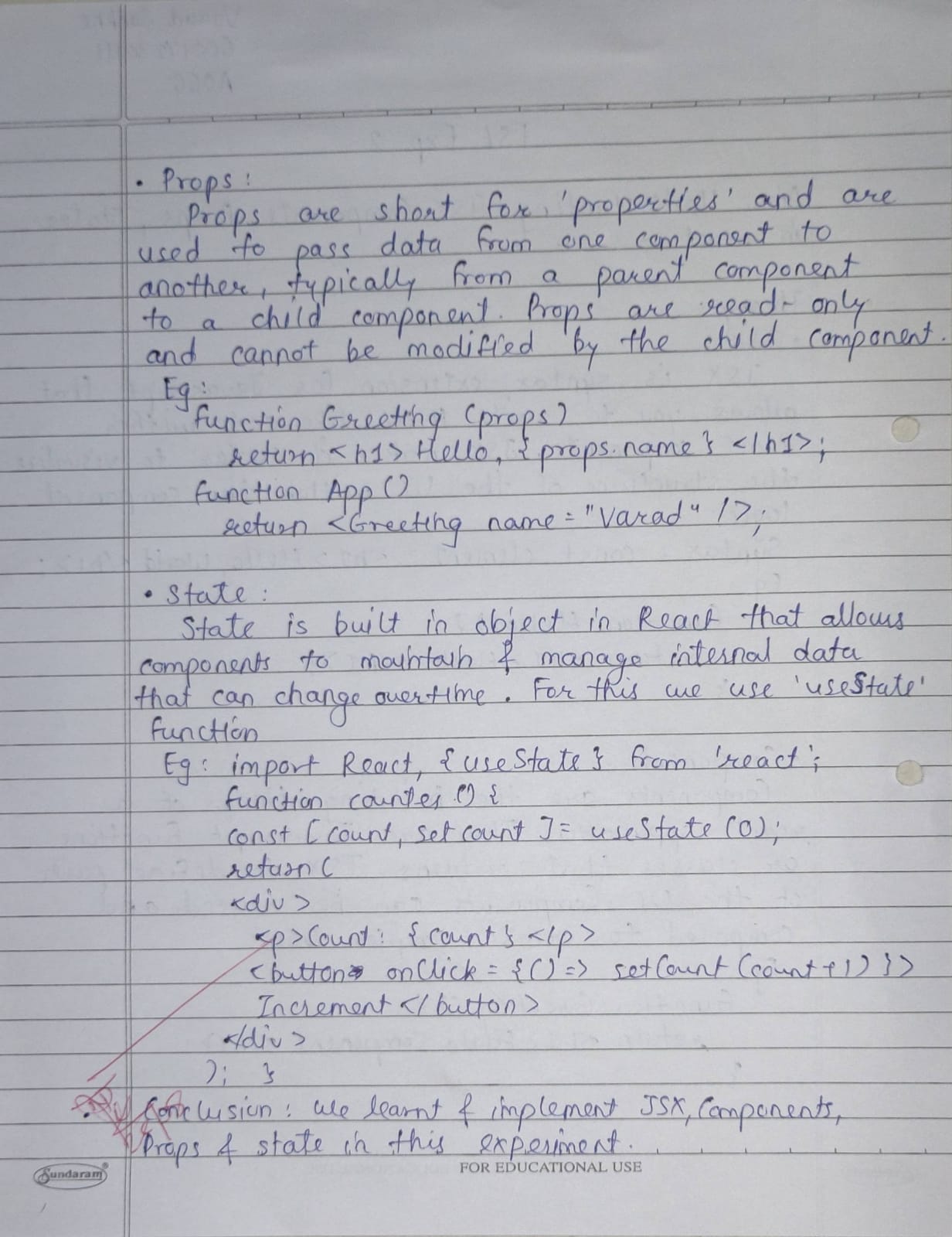
**Experiment 2**

**Name: Varad Sankhe SAP ID: 60017230111**

****

****

| **Date:** | **6/8/2024** |
| --- | --- |
| Aim | **Understanding JSX, Components, Props, State in React.** |
| Software | React,Node.js |
| Pre-requisite | Active internet connection |
| Theory | Understanding JSX What is JSX?  JSX stands for JavaScript XML.  JSX allows us to write HTML in React.  JSX makes it easier to write and add HTML in React.  Coding JSX  JSX allows us to write HTML elements in JavaScript and place them in the DOM without any createElement()  and/or appendChild() methods.  JSX converts HTML tags into react elements.  You are not required to use JSX, but JSX makes it easier to write React applications.  Here are two examples. The first uses JSX and the second does not:   Note: that in order to embed a JavaScript expression inside JSX, the JavaScript must be wrapped with curly braces, {}.Exercise for JSX:1) Example 1: Texerc2) Example 2: In this example where conditional expression is embedded in JSX3) Example 3: In this example we have wrapped h1, h2, and h3 tags under a single div element and rendered them to HTML4) Example 4: Converting HTML to JSXComponents and Props Components let you split the UI into independent, reusable pieces, and think about each piece in isolation. This page provides an introduction to the idea of components. You can find a [detailed component API reference here](https://legacy.reactjs.org/docs/react-component.html).  Conceptually, components are like JavaScript functions. They accept arbitrary inputs (called “props”) and return React elements describing what should appear on the screen.               Exercise:1) Example: Create a function component called welcome.2) Example: Create a class component called welcome.3) Example: Renders a component named Welcome to the Screen.4) Example: Create the list of unordered elements, where we will dynamically insert StudentName for every object from the data array.React PropsProps are arguments passed into React components.Props are passed to components via HTML attributes.If you have a variable to send, and not a string as in the example above, you just put the variable name inside curly brackets:  Or if it was an object:Note: React Props are read-only! You will get an error if you try to change their value.State React components has a built-in state object.  The state object is where you store property values that belong to the component.  When the state object changes, the component re-renders. |
| Code | App.jsx import React, { useState } from 'react';  import BgChanger from './components/BgChanger';  import Bgdisplay from './components/Bgdisplay';  function App() {  const [bgColor, setBgColor] = useState('white');  const changeBackground = (color) => {  setBgColor(color);  };  return (  <div  className="min-h-screen flex flex-col justify-center items-center"  style={{ backgroundColor: bgColor }} // Use inline style for dynamic background color  >  <Bgdisplay bgColor={bgColor} />  <BgChanger changeBackground={changeBackground} />  </div>  );  }  export default App;  import React from 'react';  const colors = ['red', 'blue', 'green', 'yellow', 'purple', 'pink', 'orange'];  const BgChanger = ({ changeBackground }) => {  return (  <div className='fixed bottom-5 flex gap-4'> {/\* Corrected 'botton-5' to 'bottom-5' \*/}  {colors.map((color) => (  <button  key={color}  onClick={() => changeBackground(color)}  className="text-white font-bold py-2 px-4 rounded border" // Corrected 'border-red-500' to 'border-red-500 border'  style={{ backgroundColor: color }} // Inline style for button background color  >  {color}  </button>  ))}  </div>  );  }  export default BgChanger;  import React from 'react';  const Bgdisplay = ({ bgColor }) => {  return (  <div className='text-3xl font-bold'>  Background color is {bgColor.charAt(0).toUpperCase() + bgColor.slice(1)}  </div>  );  }  export default Bgdisplay; |
| Result |  |
| Conclusion | In this experiment, we learned to use JSX for writing HTML in JavaScript, created and managed React components, and understood props and state. JSX simplifies UI creation, components help build reusable UI elements, props pass data to components, and state manages dynamic data within components. |