**Task 3**

**Title:** Context

**Description:** Create a component that uses React context for global state management.

**Project Overview:**

The application is designed to use of React Context enables a seamless theme management experience, enhancing user interaction and accessibility. It showcase a personal portfolio with dark and light mode managed by context management.

**Key Features:**

* **Theme Toggle**: Users can switch between light and dark themes with a single button click.
* **Responsive Design**: The layout adapts to various screen sizes, providing a consistent user experience across devices.

**Technical Details:**

* **Technologies Used**:
  + React for building the user interface.
  + React Icons for social media icons.
  + Tailwind CSS for styling and layout management.
* **State Management**: Utilizes React Context API to manage the theme state globally across the application.
* **Component Structure**: The application consists of a main **App** component wrapped in a **ThemeProvider** to provide the theme context.

**Code Structure:**

* **App**: The main application component that renders the UI and handles theme toggling.
* **ToggleBtn**: A button component for toggling between themes.
* **ThemeContext**: Provides the context for theme management.
* **ThemeProvider**: Wraps the application to supply the theme context.

**Example Code Snippet:**

import React, { createContext, useContext, useState } from 'react';

const ThemeContext = createContext();

const ThemeProvider = ({ children }) => {

  const [theme, setTheme] = useState('light');

  const toggleTheme = () => {

    setTheme((prevTheme) => (prevTheme === 'light' ? 'dark' : 'light'));

  };

  return (

    <ThemeContext.Provider value={{ theme, toggleTheme }}>

      {children}

    </ThemeContext.Provider>

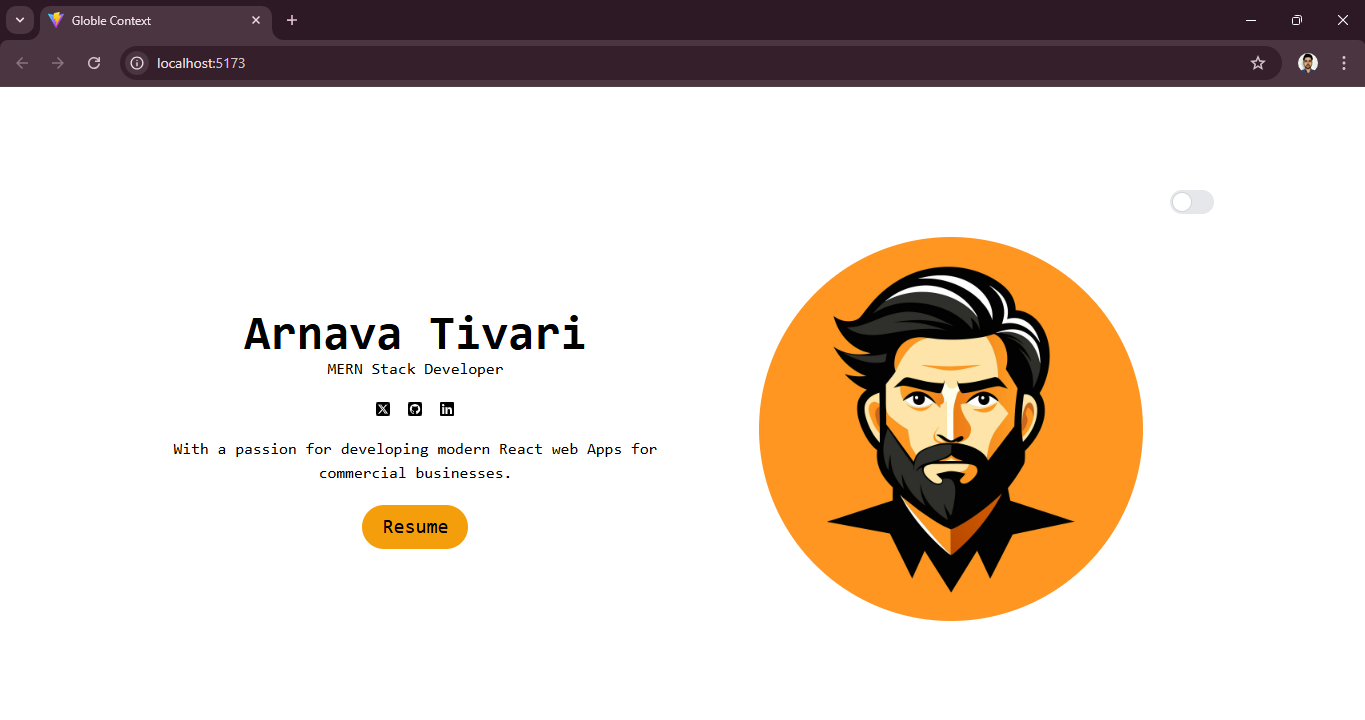
  );

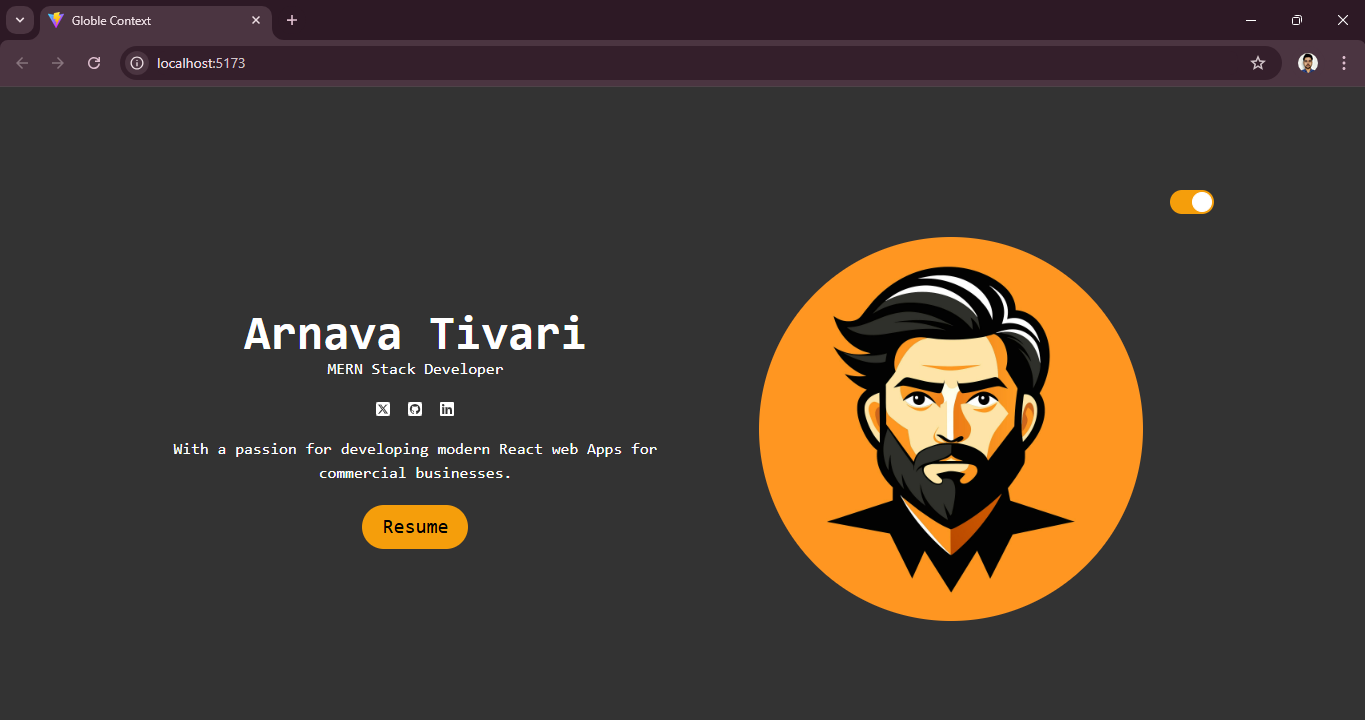
};

const useTheme = () => useContext(ThemeContext);    // Custom hook for using the theme context

export { ThemeContext, ThemeProvider, useTheme };

**Screenshots:**

****

****

**Conclusion:**

This project successfully demonstrates the use of React Context for global state management, specifically for theme toggling. The implementation of React Context not only simplifies state management but also enhances the overall user experience.