Scenario: Student Marks Card (Using Props)

Name: B. Rithwik

HT No: 2303A52330

Batch: 35

Scenario Description

In this assignment, a React-based Student Marks Card application is developed to demonstrate the concept of passing data from a parent component to child components using props. The application clearly illustrates how React follows a unidirectional data flow, where the parent component manages the data and the child component receives and uses it.

Component Structure

The application consists of two main components:

1. App Component (Parent Component):
   - Acts as the root component of the application.
   - Stores all student-related data such as name, roll number, and marks.
   - Passes this data to the child component using props.

2. StudentCard Component (Child Component):
   - Receives student data from the App component using props.
   - Displays the received data in a structured format.
   - Calculates total marks and grade for each student.

Implementation Tasks and Explanation

1. Storing Student Details in App Component:
   The App component stores student information in an array of objects. Each object contains the student's name, roll number, and an array of marks. This ensures centralized data management in the parent component.

2. Passing Data Using Props:
   The App component uses props to pass student details to the StudentCard component. Props provide a secure and structured way to transfer data from parent to child components.

3. Displaying Data in the Child Component:
   The StudentCard component receives the data through props and displays the student's name, roll number, and subject-wise marks dynamically.

Bonus Challenge Explanation

1. Calculating Total and Grade in Child Component:
   The StudentCard component calculates the total marks by summing the marks array. Based on

the total marks, a grade is assigned using conditional statements. This logic is intentionally placed in the child component to demonstrate how props can be processed inside child components.

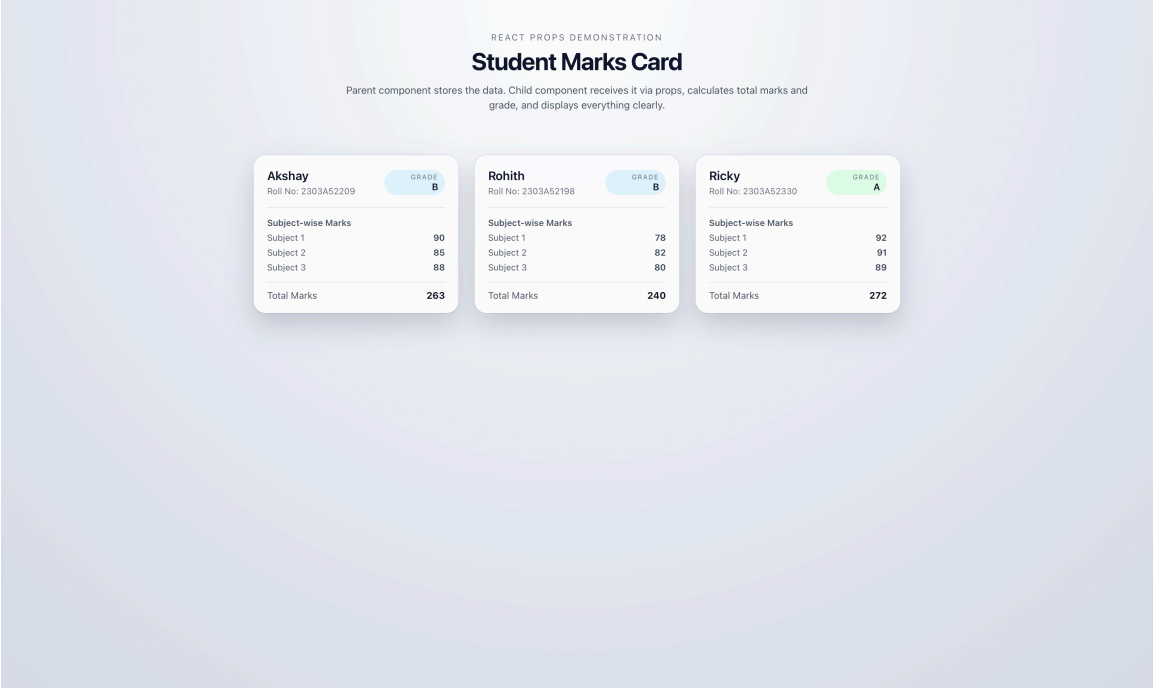
2. Reusing the Component for Multiple Students:
The same StudentCard component is reused multiple times by iterating over the students array using the map() method. This approach follows the DRY (Don't Repeat Yourself) principle and enhances reusability.

Working of the Application

When the application is executed using the npm start command, the App component renders multiple StudentCard components. Each StudentCard displays complete student information, including calculated total marks and grade. This confirms that props are working correctly and that the parent–child communication is successfully implemented.

Expected Output

• Student data is displayed correctly using props.
• Clear parent–child data flow is demonstrated.
• Reusable StudentCard components are used for multiple students.
• Total marks and grades are calculated accurately in the child component.



Conclusion

This assignment successfully demonstrates the core React concept of props and parent–child communication. By separating data storage and presentation logic, the application follows best practices in React development and ensures clean, maintainable, and reusable code.