Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment Title: Student application by using angular: implement Student Grade and all academic details by using angular

Aim/Objective

To develop a Student Application using Angular that manages student grades and academic details through a structured component-based architecture, incorporating data binding, services, and routing to create a dynamic and interactive user experience.

Description:

This project involves creating a web-based Student Application using Angular that allows users to manage and view student academic details, including grades, subjects, and personal information. The application utilizes Angular features such as components, services, routing, and two-way data binding to build a responsive and scalable user interface. It demonstrates how Angular can be used for real-time data handling and dynamic UI updates in an academic management system.

Prerequisites:

- Basic knowledge of HTML, CSS, and JavaScript.
- Understanding of TypeScript fundamentals, as Angular is built on it.
- Familiarity with Angular framework basics – components, modules, and templates.
- Knowledge of Angular CLI for project setup and command execution.
- Experience with two-way data binding, directives, and pipes.
- Awareness of Angular services and dependency injection.
- Understanding of routing and navigation in Angular applications.
- Working environment with Node.js, npm, and Angular CLI installed.

Pre-Lab: Before starting the lab activity, students should:

- Install Angular CLI and set up the development environment using npm.
- Create a new Angular project using ng new and understand its folder structure.
- Review key Angular concepts like components, modules, and services.
- Understand how to use two-way data binding with [(ngModel)].
- Learn how to define models (interfaces/classes) to structure student data.
- Plan the component hierarchy for features like student form, grade display, and academic detail view

In-Lab: During the lab session, students will:

- Create components such as:
    - student-form – to input student information
    - grade-list – to display grades
    - academic-details – to show full academic info
- Use Angular services to manage and share student data between components.
- Implement two-way data binding to update data in real-time.
- Use Angular directives and pipes to format and filter student data (e.g., GPA rounding, pass/fail status).
- Create and navigate routes using Angular Router for a multi-page experience.
- Apply styling using CSS or Angular Material for a clean UI.
- Test the application to ensure data is being added, displayed, and navigated correctly.
- Debug and document the application with meaningful comments and structure.

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Procedure/Program:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Data and Results:

Analysis and Inferences:

Experiment #	<TO BE FILLED BY STUDENT>	Student ID	<TO BE FILLED BY STUDENT>
Date	<TO BE FILLED BY STUDENT>	Student Name	<TO BE FILLED BY STUDENT>

Sample viva voce questions:

1. **What is the role of components in Angular, and how did you use them in your student application?**

2. **Explain two-way data binding in Angular with an example.**

3. **How are services used in Angular, and why are they important for this project?**

4. **What is routing in Angular, and how did you implement navigation between pages?**

5. **What are Angular pipes, and how did you use them in your project for formatting data?**

Evaluator Remark (if Any):	**Marks Secured:_____out of 50**
	Signature of the Evaluator with Date