Software Testing Plan

1. Introduction

This document describes the testing strategy, objectives, and scope for the **Food Delivery & Table Reservation Application**. The purpose of testing is to ensure that the system functions correctly, meets user requirements, and is free from critical defects before deployment.

2. Test Strategy

The following testing approaches will be used:

- **Unit Testing:**
  Individual modules such as login, menu display, order placement, payment, and reservation will be tested separately.
- **Integration Testing:**
  Interaction between modules such as order placement with payment processing and reservation with admin confirmation will be tested.
- **System Testing:**
  The complete system will be tested to ensure all functionalities work as expected in an integrated environment.
- **Acceptance Testing:**
  The system will be tested from the end-user perspective to verify that it meets customer and admin requirements.

3. Test Environment

The test environment specifies the resources required for testing.

- **Hardware:**
  Laptop / Desktop / Mobile device with minimum 4GB RAM
- **Software:**
  - Operating System: Windows / Linux
  - Browser: Chrome / Edge
  - Backend Server: NodeJs
  - Database: MongoDB
- **Test Data:** Sample user accounts, menu items, orders, payments, and reservation data.

4. Test Cases

Test Case	Input Data	Expected Output	Status
Login	Valid username & password	Successful login	Pass
Menu View	User selects menu	Menu items displayed	Pass
Order Placement	Select food items	Order created successfully	Pass
Payment	Valid payment details	Payment successful	Pass
Reservation	Date & time selected	Reservation confirmed	Pass

5. Defect Management

- All defects identified during testing will be logged.
- Each defect will be categorized based on severity.
- Developers will fix the reported defects.
- Retesting will be performed to ensure defect resolution.
- A final verification will confirm system stability.