Sprint 3 Release & Final Sprint Requirements List

1. Sprint 3 Release Summary

The primary objective of the combined Sprints 1 and 2 was to deliver a complete, functional, and modern GUI-based Grade Calculator with essential persistence and advanced analytical features.

Achievements to Date:

- **Grade Logic Engine Complete:** All core calculation logic (weighted average, category scores, letter grades) is fully implemented.
- **Advanced Prediction Complete:** The core innovative feature—calculating the required score needed to achieve a target final grade—is functional.
- **Modern GUI Ready:** The application is built using PyQt5, providing a responsive and modern desktop interface.
- **Data Persistence:** Local saving and loading of multiple class profiles has been implemented, fulfilling the core data persistence requirement (Req 7, 8).

Scope Adjustment: Due to time constraints, the team has decided to scope down the project from a Cloud-enabled application to a fully functional and robust **Local Desktop Application**. This means all cloud, authentication, and network security requirements are deferred or cancelled (Req 13, 14, 15, 20).

2. Sprint 3 Release Backlog (Completed/Shipped)

These are the Requirements IDs from the initial stack that have been successfully developed and released up to this point.

Req ID	Description	Status	Story Points
1	User can create a new class profile.	DONE	2
2	User can define a grading category and assign a percentage weight.	DONE	3
3	**Validation:** Ensure that the sum of all category weights equals 100%.	DONE	5

4	User can add a new assignment to a category.	DONE	3
5	**Core Calculation:** Display the current overall weighted percentage grade.	DONE	5
6	Display the percentage and letter grade for each individual grading category.	DONE	3
7	**Data Persistence (Local):** Implement logic to save class and grade data to local storage.	DONE	8
8	**Data Persistence (Local):** Implement logic to load saved class and grade data.	DONE	5
9	**What-If Analysis:** User can input a hypothetical score for an ungraded assignment.	DONE	13
10	**Final Grade Prediction:** Calculate the minimum required score on a remaining assignment to achieve a user-specified target grade.	DONE	13
12	Implement the application's modern, responsive graphical user interface (GUI) using a chosen framework (PyQt5).	DONE	8
16	**Custom Grade Scale:** Allow the user to define and save their own letter grade cutoffs.	DONE	5
17	**Error Handling:** Gracefully handle the division-by-zero scenario.	DONE	2
18	**Validation:** Implement input checks to prevent non-numeric or negative values in score fields.	DONE	2
19	Implement basic unit tests for the core grade calculation logic.	DONE	3
23	**Accessibility:** Ensure the final GUI design meets WCAG minimum accessibility standards (e.g., keyboard navigation).	DONE	5
24	Develop a short video presentation demonstrating the final application's features and architecture.	NOT STARTED	3

3. Deferred and Cancelled Requirements

The following features, originally planned for Sprints 2 and 3, have been removed from the scope of this project to ensure timely delivery of a high-quality desktop application.

Req ID	Description	Reason for Deferral
13	**Security:** Implement user authentication (sign-up/log-in).	Scope reduction; shifted focus to a single-user local tool.
14	**Cloud Storage:** Store the user's class and grade data in a cloud database (e.g., Firebase).	Scope reduction; local JSON file persistence is sufficient for the MVP.
15	Implement password hashing.	Dependent on Req 13 (User Authentication), therefore cancelled.
20	**Security:** Ensure all data transmission is encrypted (HTTPS/SSL).	Not applicable since cloud storage and user authentication were cancelled.