Guidelines for Project Implementation and Submission

1. General Guidelines

- All projects must be implemented on the STM32F446RE Nucleo board.
- Work should follow an Agile approach: break requirements into user stories, plan sprints, and deliver working increments.
- Document design, implementation, testing, and Agile process in the report.
- Submit:
    1. Source code (well-commented, modular)
    2. Circuit diagram / pin mapping
    3. Test logs (UART outputs, scope captures, screenshots)
    4. Agile artifacts (user stories, backlog, sprint notes)
    5. Final report (structured as below)

2. Agile Requirements

- User Stories: Write features in the format:
  "As a developer, I want [feature] so that [benefit]."
- Acceptance Criteria: Define measurable conditions for each story.

3. Report Structure

I. Project title, student name(s), college, date

II. Short summary of project goals and outcomes

III. Agile Process Documentation

User Story ID	User Story	Action Item	Acceptance Criteria
US1	As a developer, I want [feature] so that [benefit].		List measurable conditions
US2	...		...

IV. System Design- Block diagram, hardware components, pin mapping

V. Implementation

- Key configurations (timers, ADC, UART, PWM, etc.)
- Calculations done if any
- Flowcharts or pseudocode
- Code structure explanation

VI. Testing & Results

- Logs, screenshots, oscilloscope captures
- Comparison of expected vs actual results

VII. Sprint Review and Retrospective

- Completed stories: [IDs]
- Unfinished stories: [IDs + reason]
- What went well
- What to improve

VIII. Discussion - Challenges faced and solutions

IX. Conclusion - Summary of achievements and whether success criteria were met

X. References

XI. Appendix - Full source code, extra test data

Sample Agile Plan for the simple project of interfacing an LED with pin PA5 and toggling it on each button press.

User Story ID	User Story	Action Items	Acceptance Criteria
US1	As a learner, I want the LED connected to PA5 to turn ON when I press the button so that I can see immediate feedback.	- Configure PA5 as output -Configure button pin as input with pull-up -Implement interrupt or polling logic to detect press -Turn ON LED within 100 ms	- LED turns ON within 100 ms of button press -LED state is clearly visible
US2	As a learner, I want the LED to turn OFF when I press the button again so that I can toggle it easily.	- Implement toggle logic -Maintain LED state variable -Ensure reliable state change on each press	- LED toggles OFF reliably on next press -No false toggles occur
US3	As a learner, I want the button input to be debounced so that the LED does not flicker or toggle multiple times per press.	- Implement software debounce (e.g., delay or state machine) -Test with multiple presses -Validate debounce timing	- LED toggles only once per physical press -Works consistently across 20+ presses
US4	As a learner, I want the system to initialize with the LED OFF so that I know the default state at startup.	- Set LED OFF in startup code -Initialize state variable -Ensure first press turns LED ON	- On reset/power-up, LED is OFF -First press always turns it ON
US5	As a learner, I want to log the LED state over UART so that I can verify functionality during testing.	- Initialize UART -Print “LED ON” or “LED OFF” after each toggle -Match log with LED state	- UART prints “LED ON” or “LED OFF” after each press -Logs match LED state

Sample Test Cases

Test Case ID	Test Description	Expected Result	Validation Method
TC1	Verify LED turns ON within 100 ms of button press	LED turns ON within 100 ms LED visibly ON	Visual confirmation Oscilloscope/timer
TC2	Verify LED toggles OFF on second press	LED turns OFF reliably No flicker or false toggle	Visual confirmation State variable check
TC3	Verify debounce prevents multiple toggles per press	LED toggles once per press No flicker or bounce	Visual confirmation UART log consistency
TC4	Verify LED is OFF at startup	LED is OFF at startup First press turns it ON	Visual confirmation Initial state variable
TC5	Verify UART logs match LED state	UART prints “LED ON” or “LED OFF” correctly Matches LED state	UART terminal log Cross-check with LED