TASK 3A – FEEDBACK PLAN (DISTINCTION-LEVEL TEMPLATE)

1. Introduction

In this section you should briefly explain:
• The overall purpose of gathering feedback on your system.
• Which parts of the system will be evaluated (e.g. registration, login, dashboard, bookings, UI).
• How this links back to the client and user requirements from Task 1 and Task 2.

2. Target Audiences

Clearly identify who you will collect feedback from and why they are appropriate.
Non-technical users:
• Who they are (e.g. typical users, age range, experience level).
• Why they are selected (e.g. represent real users of the system).
• What type of feedback they will give (usability, clarity, ease of use, visuals).

Technical users:
• Who they are (e.g. developers, teachers, technically confident peers).
• Why they are selected (e.g. can comment on code quality, security, structure).
• What type of feedback they will give (code quality, security, performance, scalability).

3. Feedback Methods

Explain exactly how you will collect feedback from each audience and justify your choices.
Non-technical users (end users):
• Observation while they complete set tasks – explain why this helps you see real behaviour.
• User testing tasks with a script – describe why task-based testing is suitable.
• Questionnaire with a mix of closed (ratings) and open questions – explain how this gives both numerical and detailed feedback.

Technical users:
• Code review session – justify why this is appropriate for reviewing structure, security and maintainability.
• Technical questionnaire – describe what kind of questions you will ask (e.g. about security, database design, validation).

4. Ethical, GDPR & Safeguarding Considerations

In this section you must show awareness of legal and ethical responsibilities when collecting feedback.
• Consent and confidentiality – how you will inform participants and gain their consent.
• Protecting personal data – avoiding collection of unnecessary personal details.
• Anonymous recording – how you will avoid using real names in your final write-up.
• Safeguarding if minors are involved – making sure testing is supervised and appropriate.
• Secure storage of data – briefly explain how notes or screenshots will be stored during the project.

5. User Testing Tasks (Observation Tasks)

Here you define the specific tasks non-technical users will perform. For each task you should include:
• Task description (e.g. “Register a new account”, “Log in and find your bookings”).
• Rationale – why this task is important to test.
• Expected behaviour – what a successful attempt looks like.
• What will be recorded – time taken, number of errors, help requested, comments.
• Success criteria – how you will decide if the task passed or failed.

6. Feedback Materials (Appendix Templates)

List or reference any supporting materials you will design, for example:
• Non-technical user questionnaire (with rating scales and open questions).
• Technical questionnaire for developers or teachers.
• Observation sheet template for recording user behaviour.
• Code review form to structure technical feedback.

7. Data Collection & Recording Strategy

Explain how you will store, organise and analyse the feedback you collect.
• Where the data will be stored (e.g. table in your document, spreadsheet).
• How feedback will be categorised (e.g. usability, performance, accessibility, security).
• How you will analyse quantitative data (e.g. averages, common scores).
• How you will analyse qualitative data (grouping similar comments into themes).
• How anonymity will be preserved in your final write-up.

8. Timeline

Provide a simple, realistic timeline for your feedback activities, for example:
• Phase 1 – Preparation (designing questionnaires, observation sheets).
• Phase 2 – Conducting feedback sessions with users and technical reviewers.
• Phase 3 – Analysing results and identifying improvements.

9. Improvement Strategy

Describe how you will use the feedback to improve the system.
• How you will prioritise issues (e.g. critical bugs first, then usability, then visual polish).
• How improvements will be grouped into future versions or sprints.
• How suggestions from different audiences will be balanced.
• How this links back to meeting the original requirements and success criteria.

10. Conclusion

End with a short paragraph explaining why your feedback plan is effective and how it will help make the final product more usable, reliable, secure and aligned with user and client needs.