

Sprint 3 Presentation Guidelines, Deliverables, and Rubrics

Objective

Sprint 3 focuses on the final delivery of a fully functional, live-hosted application with complete core features, advanced integrations, performance improvements, and robust deployment practices. Emphasis is placed on validating production-readiness, API integration, and UI/UX refinement.

Presentation Structure, Deliverables, and Evaluation Steps

1. Recap and Sprint 3 Goals (5%)

Deliverables to Check:

- ✓ Slide summarizing Sprint 1 & 2 key achievements
- ✓ Description of feedback from previous sprints and how they were addressed
- ✓ Slide outlining Sprint 3 goals and intended outcomes
- ✓ Reflection on team improvement and project direction

Technical Verification Steps:

1. Open the presentation slides and look for a dedicated recap section
2. Check if goals are clearly outlined for Sprint 3
3. Look for reflective insights related to feedback application

2. Technology Stack & Architecture Updates (15%)

Deliverables to Check:

- ✓ Updated architecture diagram
- ✓ List of tools, frameworks, and libraries used
- ✓ Explicit statement confirming Angular is NOT used
- ✓ Frontend framework evidence in package.json
- ✓ Source code inspection for Angular traces
- ✓ Browser console check with window.ng

Technical Verification Steps:

1. Open architecture diagram for clarity and completeness
2. Check the package.json for frontend stack used (React, Vue, etc.)
3. Search the repository for 'angular.json', '@angular', or 'ng' scripts
4. Open the live app and run 'window.ng' in browser console — should return undefined

3. Core Functionalities & Feature Completion (25%)

Deliverables to Check:

- ✓ Live demo of key user and admin features

- ✓ Form validation and error handling
- ✓ Working navigation and session management
- ✓ Well-structured codebase with modular services

Technical Verification Steps:

1. Interact with live demo or local app build and test core features
2. Submit sample forms to trigger validations
3. Inspect localStorage or cookies for session tokens
4. Review the repository for modular file structure

4. API Integration and Validation (20%)

Deliverables to Check:

- ✓ DevTools/Network tab showing API requests
- ✓ Postman collection for API testing
- ✓ Screenshots of error handling
- ✓ Fallback logic demonstration
- ✓ API service code with clear structure and comments

Technical Verification Steps:

1. Open browser DevTools > Network tab, confirm API activity
2. Run Postman collection provided in submission
3. Trigger known error conditions and inspect system response
4. Review error logs in the browser or terminal
5. Check repo for files like apiService.js or api.js with documented functions

5. Hosting and Live Deployment (15%)

Deliverables to Check:

- ✓ Live site URL on platforms like Vercel, Firebase, etc.
- ✓ Responsive layout across devices
- ✓ .env.example file included in repo
- ✓ Deployment steps in README

Technical Verification Steps:

1. Open and navigate the live site on mobile and desktop
2. Check console for functional API on the live site
3. Check for .env.example and verify deployment steps in README

6. Testing, Performance & Bug Fixes (10%)

Deliverables to Check:

- ✓ Performance benchmarks (response/load time)
- ✓ Bug fix list since Sprint 2
- ✓ Unit test and E2E test results
- ✓ Error logs or issue trackers

Technical Verification Steps:

1. Open test report files or screenshots
2. Use Lighthouse for performance audits
3. View GitHub Issues or Trello for bug history

7. Presentation & Team Reflection (10%)

Deliverables to Check:

- ✓ Clean, organized slides
- ✓ Team coordination and participation
- ✓ Insights on challenges and learnings
- ✓ Effective time use and Q&A handling

Technical Verification Steps:

1. Observe the team presentation
2. Check slide design and clarity
3. Evaluate individual member contributions

Numeric Rating Criteria

Each section is scored based on a 10-point scale. Use the technical validation steps provided below each category to determine a fair score. Use the following general scale for guidance:

10-9: Excellent — fully implemented, well-documented, professionally executed

8-7: Good — mostly complete, minor gaps or inconsistencies

6-5: Needs Improvement — partially complete, lacks clarity or robustness

4-0: Poor — incomplete, broken, or missing features

Name of Application:

Team Name:

Evaluator's Name and Team Name:

Detailed Criteria with Technical Validation and Scoring

Recap and Sprint 3 Goals (5%)

Deliverables to Check:

- Slide summarizing Sprint 1 & 2 key achievements
- Description of feedback from previous sprints and how they were addressed
- Slide outlining Sprint 3 goals and intended outcomes
- Reflection on team improvement and project direction

Technical Validation Steps:

- Open the presentation slides and look for a dedicated recap section
- Check if goals are clearly outlined for Sprint 3
- Look for reflective insights related to feedback application

Score (0-10): _____

Evaluator Notes/Recommendations:

Technology Stack & Architecture Updates (15%)

Deliverables to Check:

- Updated architecture diagram
- List of tools, frameworks, and libraries used
- Explicit statement confirming Angular is NOT used
- Frontend framework evidence in package.json
- Source code inspection for Angular traces
- Browser console check with window.ng

Technical Validation Steps:

- Open architecture diagram for clarity and completeness
- Check the package.json for frontend stack used (React, Vue, etc.)
- Search the repository for 'angular.json', '@angular', or 'ng' scripts
- Open the live app and run 'window.ng' in browser console — should return undefined

Score (0-10): _____

Evaluator Notes/Recommendations:

Core Functionalities & Feature Completion (25%)

Deliverables to Check:

- Live demo of key user and admin features
- Form validation and error handling

- Working navigation and session management
- Well-structured codebase with modular services

Technical Validation Steps:

- Interact with live demo or local app build and test core features
- Submit sample forms to trigger validations
- Inspect localStorage or cookies for session tokens
- Review the repository for modular file structure

Score (0-10): _____

Evaluator Notes/Recommendations:

API Integration and Validation (20%)

Deliverables to Check:

- DevTools/Network tab showing API requests
- Postman collection for API testing
- Screenshots of error handling
- Fallback logic demonstration
- API service code with clear structure and comments

Technical Validation Steps:

- Open browser DevTools > Network tab, confirm API activity
- Run Postman collection provided in submission
- Trigger known error conditions and inspect system response
- Review error logs in the browser or terminal
- Check repo for files like apiService.js or api.js with documented functions

Score (0-10): _____

Evaluator Notes/Recommendations:

Hosting and Live Deployment (15%)

Deliverables to Check:

- Live site URL on platforms like Vercel, Firebase, etc.
- Responsive layout across devices
- .env.example file included in repo
- Deployment steps in README

Technical Validation Steps:

- Open and navigate the live site on mobile and desktop
- Check console for functional API on the live site
- Check for .env.example and verify deployment steps in README

Score (0-10): _____

Evaluator Notes/Recommendations:

Testing, Performance & Bug Fixes (10%)

Deliverables to Check:

- Performance benchmarks (response/load time)
- Bug fix list since Sprint 2
- Unit test and E2E test results
- Error logs or issue trackers

Technical Validation Steps:

- Open test report files or screenshots
- Use Lighthouse for performance audits
- -View GitHub Issues or Trello for bug history

Score (0-10): _____

Evaluator Notes/Recommendations:

Presentation & Team Reflection (10%)

Deliverables to Check:

- Clean, organized slides
- Team coordination and participation
- Insights on challenges and learnings
- Effective time use and Q&A handling

Technical Validation Steps:

- Observe the team presentation
- Check slide design and clarity
- Evaluate individual member contributions

Score (0-10): _____

Evaluator Notes: