

Judging Template

≡ Category	Workshop
≡ Content Type	Rubric
≡ Workshop Day	Competition

Judging Overview

Purpose:

Provide a consistent framework for evaluating each team's final application, engineering process, adaptability, and presentation.

Total Score: 100 points

Judges: TBD

Judging Time Per Team: TBD

Judging Principles

Judges should evaluate both the final product and the engineering process used to create it.

A visually impressive website should not automatically win if the underlying functionality is incomplete.

Likewise, technical complexity should not automatically receive a higher score if it does not improve the solution.

Judges should consider:

- Does it work?
- Does it satisfy the requirements?
- Is the application well designed?
- Can the team explain how it works?
- Did the team demonstrate effective problem solving?
- Did the team adapt successfully to the surprise change?
- Did the team use AI thoughtfully rather than blindly?

- Is the final application production ready?
-

Scoring Summary

Category	Points
Functional Requirements	25
Technical Implementation	20
UI/UX	15
Problem Solving and AI Usage	15
Surprise Change	10
Production Quality	10
Final Presentation	5
Total	100

1. Functional Requirements

Maximum: 25 points

Evaluate whether the application successfully solves the challenge and satisfies the required functionality.

Consider

- Core features work correctly
- Authentication works
- Persistent data works
- User-specific behavior works
- Required user flows work
- Application behaves correctly after refresh
- Challenge requirements are satisfied
- Application can be demonstrated without major failures

Score

Score: ____ / 25

Judge Notes

TBD

2. Technical Implementation

Maximum: 20 points

Evaluate the quality and understanding of the technical implementation.

Consider

- Application architecture makes sense
- Excalidraw diagram reflects the implemented system
- Firebase is integrated appropriately
- Authentication and data responsibilities are understood
- Git and GitHub were used throughout development
- Repository contains meaningful development history
- Code structure is understandable
- Team can explain important implementation decisions
- Team understands the code produced with AI assistance

Score

Score: ____ / 20

Judge Notes

TBD

3. UI/UX

Maximum: 15 points

Evaluate the usability and visual quality of the application.

Consider

- Interface is clear and understandable
- Navigation is intuitive

- Visual design is consistent
- Important actions are easy to find
- User feedback is provided when appropriate
- Error and loading states are handled reasonably
- Application is responsive
- Basic accessibility has been considered
- Final implementation is reasonably consistent with the team's design process

Score

Score: ____ / 15

Judge Notes

TBD

4. Problem Solving and AI Usage

Maximum: 15 points

Evaluate how effectively the team used Codex, Claude Code, and their own reasoning during development.

The objective is not to reward the team that generated the most code with AI.

The objective is to evaluate whether the team used AI effectively as an engineering assistant.

Ask the Team

What was one difficult technical problem you encountered?

TBD

How did you investigate it?

TBD

How did Codex or Claude Code help?

TBD

Did AI ever give you an incorrect or incomplete solution?

TBD

How did you recognize and correct it?

TBD

How did you verify AI-generated code?

TBD

Did you use any hooks or workflow automation? If so, why?

TBD

Consider

- Team can describe a real debugging process
- Team collected evidence before making changes
- AI was given useful context
- AI-generated code was tested
- Team recognized incorrect AI output when applicable
- Team understands the implementation
- Hooks or automation were used purposefully when applicable

Score

Score: ____ / 15

Judge Notes

TBD

5. Surprise Change

Maximum: 10 points

Evaluate how effectively the team adapted when the additional requirement was announced.

Consider

- Team correctly understood the new requirement
- Existing architecture was adapted appropriately

- Team prioritized effectively
- Existing functionality was preserved
- New functionality works
- Team tested the change
- Team can explain how they approached the change

Ask the Team

What did you have to change after the new requirement was announced?

TBD

Did the change affect your original architecture?

TBD

What did you prioritize first?

TBD

Score

Score: ____ / 10

Judge Notes

TBD

6. Production Quality

Maximum: 10 points

Evaluate whether the application was successfully taken from development to production.

Verify

- ☐ Application is deployed on Vercel
- ☐ Production URL works
- ☐ Custom domain works
- ☐ Cloudflare DNS is configured
- ☐ HTTPS works

- ☐ Authentication works in production
- ☐ Persistent data works in production
- ☐ Important routes work
- ☐ No obvious critical console errors
- ☐ Basic search engine discovery requirements were considered
- ☐ PageSpeed Insights was used

Ask the Team

What did PageSpeed Insights identify?

TBD

What did you improve based on the results?

TBD

Score

Score: ____ / 10

Judge Notes

TBD

7. Final Presentation

Maximum: 5 points

Evaluate how effectively the team communicates its work.

Consider

- Problem is explained clearly
- Solution is explained clearly
- Demonstration is organized
- Important functionality is shown
- Architecture can be explained
- Technical decisions can be explained
- Team responds effectively to judge questions

Score

Score: ____ / 5

Judge Notes

TBD

Technical Verification

Judges or technical instructors may use this section to verify functionality directly.

Authentication

- ☐ Sign-up works
- ☐ Sign-in works
- ☐ Sign-out works
- ☐ Authentication state works

Data

- ☐ Data can be created
- ☐ Data can be retrieved
- ☐ Data persists after refresh
- ☐ User-specific data behaves correctly

Production

- ☐ Vercel deployment works
- ☐ Custom domain works
- ☐ HTTPS works
- ☐ Firebase works in production

Repository

- ☐ GitHub repository is accessible
- ☐ Repository contains meaningful commits

- ☐ Project can be understood from the repository
- ☐ No obvious sensitive credentials are committed

Architecture

- ☐ Excalidraw diagram was submitted
 - ☐ Diagram represents major application interactions
 - ☐ Team can explain the diagram
 - ☐ Implementation reasonably reflects the planned architecture
-

Judge Questions

Use these when additional clarification is needed.

Architecture

Walk us through your Excalidraw diagram. What happens from the moment a user performs an action until the data is stored or retrieved?

Technical Decisions

What technical decision did your team spend the most time thinking about, and why?

Debugging

Show us or describe one bug that took meaningful investigation to solve.

AI

Give us an example where Codex or Claude Code helped you solve a problem rather than simply generate code.

Verification

How did you determine that the AI-generated implementation was actually correct?

Adaptability

| How did the surprise requirement affect your original implementation?

Production

| What changed when you moved the application from localhost to production?

Improvement

| If you had another day to work on this application, what would you improve first and why?

Penalties

Use only when applicable and agreed upon by organizers before judging begins.

Missing Required Deliverable

Penalty: TBD

Application Cannot Be Accessed

Penalty: TBD

Repository Cannot Be Accessed

Penalty: TBD

Competition Rule Violation

Penalty: TBD

Other

Reason: TBD

Penalty: TBD

Bonus Points

Recommended default:

No bonus points.

All teams should be evaluated using the same 100-point rubric.

If bonus points are introduced, they must be defined before the competition begins.

Final Score

Category	Score
Functional Requirements	___ / 25
Technical Implementation	___ / 20
UI/UX	___ / 15
Problem Solving and AI Usage	___ / 15
Surprise Change	___ / 10
Production Quality	___ / 10
Final Presentation	___ / 5
Penalties	___
Final Score	___ / 100

Judge Summary

Strongest Aspect

TBD

Biggest Weakness

TBD

Technical Assessment

TBD

Overall Comments

TBD

Award Consideration

Would you consider this team for a winning position?

- ☐ Yes
- ☐ Maybe
- ☐ No

Reason:

TBD

Tie-Breaker

If two teams receive the same final score, compare them in this order:

1. Functional Requirements
2. Problem Solving and AI Usage
3. Surprise Change
4. Technical Implementation
5. Production Quality

If teams remain tied, judges should discuss the overall quality of the solutions and reach a consensus.

Post-Judging Notes

Complete after judging is finished.

Scoring concerns:

TBD

Rubric improvements for future events:

TBD

Common weaknesses across teams:

TBD

Common strengths across teams:

TBD