INTERN ASSESSMENT BRIEF

Event Guest Registration App

48-Hour Prototype Challenge

CONFIDENTIAL — FOR ASSESSOR AND CANDIDATE USE ONLY

Version 1.0

Part B — Candidate Brief

Hand the following pages to the candidate. Do not share Part A or the rubric.

Background

You have been hired as a junior developer for a tech consultancy. A client has approached your team with an urgent request — they are running an event in a few weeks and need a simple Guest Registration App to be used at their registration counter on the day.

The client's only firm requirement is that guests should be able to check in using a QR code. However, if you believe there is a better or more practical approach to guest check-in for the type of event you are designing for, you are welcome to propose an alternative — provided you can clearly explain your reasoning and why it would better serve the client. Part of being a good developer is knowing when to follow a requirement and when to respectfully challenge one.

Everything else — the type of event, the technology stack, the features you build — is up to you to decide. Your manager has asked you to take the lead on this.

Your Task

Over the next 48 hours, build a working prototype of the Guest Registration App and prepare to present it to your manager.

You are free to make your own decisions about what type of event this is for, what technology to use, and what features to build. There are no wrong answers — but you will be asked to explain every decision you make.

Required: AI Coding Tools

You are required to use an AI-assisted coding harness as your primary development environment. This means working inside a tool that integrates AI directly into your coding workflow — not simply pasting code into a chatbot.

Acceptable tools include (but are not limited to):

- Claude Code (Anthropic) — terminal-based AI coding agent
- OpenCode — open-source terminal-based AI coding agent
- Cursor — AI-native code editor
- GitHub Copilot (within VS Code or JetBrains)
- Codex CLI (OpenAI)
- Windsurf — AI-native code editor

You may also use chat-based AI tools (such as ChatGPT or Claude.ai) for planning, research, or brainstorming alongside your coding harness. However, your primary development workflow should run through one of the tools above.

This is not a test of whether you can code without help. It is a test of how effectively you collaborate with AI tools — what you prompt, what you accept, what you reject, and how you iterate.

Deliverables

By the end of 48 hours, prepare the following:

1. A Working Prototype

It does not need to be complete or polished, but the core check-in flow should be demonstrable. A simple app that works end-to-end is more valuable than an ambitious one that is half-finished.

2. A Short Rationale Document

A README, one-pager, or a few slides covering:

- The assumptions you made about the event and how the app would be used
- The features you prioritised and why
- What you consciously decided to leave out
- What you would build next if given more time
- If you proposed an alternative to QR code check-in, your reasoning

3. AI Tool Documentation

Screenshots, exported chat logs, terminal session recordings, or a short written summary showing how you used AI tools during the build. This should include:

- Which AI coding harness you used and how you set it up
- At least one example of where you changed or rejected something the AI suggested, and why
- Any moments where the AI helped you learn something new or solve a problem you were stuck on

4. A 15-Minute Presentation

Walk your manager through the prototype and your decisions. Use the suggested structure below as a loose guide, not a strict script.

Suggested Presentation Structure

1. What type of event did you design for, and why?
2. What assumptions did you make about how the app would be used on the day?
3. What did you decide to build, and what did you consciously leave out?
4. If you proposed an alternative to QR code check-in, explain your reasoning.
5. Show the prototype working — walk through the core check-in flow.
6. How did you set up your coding environment and AI tools?

7. Where did AI tools help you most? Where did they fall short or mislead you?
8. If you had another 48 hours, what would you tackle next?

A Note on Scope

You are not expected to build a production-ready system.

A beginner who builds a simple, working check-in flow and can clearly explain their thinking will be rated more highly than one who attempts a complex system they cannot explain.

Focus on getting the core experience right before adding anything extra.

Good luck. We look forward to seeing what you build and, more importantly, how you think about it.