Contents

Scenario: Only 3 Engineers and 6 Months

Describe your prioritization, architecture, and delivery plan for a large project with a small team and tight deadline.

Purpose

- Tests your ability to deliver under resource and time constraints
- Evaluates your prioritization, technical decision-making, and risk management
- Assesses your focus on business value and communication

Summary

Define a clear MVP, outsource non-core components. Use BaaS (Firebase, Auth0). Optimize for delivery, not scale. Cut any non-essential metrics or features.

Requirements & Prioritization

- **Clarify Business Goals:** Work closely with stakeholders to define the most critical outcomes
- **Define MVP (Minimum Viable Product):** Identify the smallest set of features that delivers core value
- **Prioritize Ruthlessly:** Defer or cut non-essential features, metrics, and optimizations
- **Timeboxing:** Break work into short, focused sprints with clear deliverables

---

Architecture & Technical Choices

- **Leverage BaaS (Backend-as-a-Service):** Use managed services like Firebase, Auth0, or AWS Amplify for authentication, storage, and notifications
- **Outsource Non-Core Components:** Use third-party APIs or contractors for features outside your team's expertise
- **Monolithic or Modular Monolith:** Prefer simple, maintainable architectures over microservices for speed

- **Automate CI/CD:** Use simple pipelines for fast feedback and deployment

---

Delivery & Team Practices

- **Agile, Lightweight Process:** Daily standups, clear ownership, and rapid iteration
- **Code Reviews & Pairing:** Maintain quality with lightweight reviews and knowledge sharing
- **Continuous User Feedback:** Ship early, gather feedback, and iterate quickly
- **Documentation:** Keep docs minimal but up-to-date for onboarding and handoff

---

Risk Management

- **Identify Bottlenecks Early:** Address single points of failure and knowledge silos
- **Scope Creep Control:** Push back on new requirements unless critical
- **Contingency Planning:** Have backup plans for key risks (e.g., vendor lock-in, BaaS limitations)

Communication

- **Transparent Updates:** Regularly update stakeholders on progress, risks, and changes
- **Demo Early and Often:** Show working software to build trust and get feedback

---

Metrics for Success

- Time to MVP delivery
- Number of features delivered vs. planned
- User feedback and adoption
- Team velocity and morale