

Software project specification

For web, mobile, backend, payment and integration projects. Fill in what you know; the gaps become the agenda for the scoping call. The more concrete the answers, the more accurate the estimate.

1. Overview

Project name	
Your name, company, contact	
What it is	<i>Web app · mobile app · backend/API · integration · payment gateway · migration · other</i>
Problem it solves	<i>Who has the problem today and how they handle it now</i>
Success looks like	<i>Three measurable outcomes six months after launch</i>
Deadline and budget band	

2. Users and roles

User types	<i>e.g. customer, merchant, admin, support, partner — and roughly how many of each</i>
What each role must be able to do	<i>Key actions per role</i>
Authentication	<i>Email/password, OTP, SSO, 2FA, device binding; existing identity provider?</i>

3. Scope

MUST HAVE (LAUNCH)	NICE TO HAVE (LATER)
EXPLICITLY OUT OF SCOPE <i>What we should not build, so it is not assumed.</i>	

4. Platforms and integrations

Platforms	<i>Web (browsers) · Android · iOS · desktop; minimum versions</i>
Existing systems to integrate	<i>CRM, billing, ERP, trading server, bank/payment gateways, messaging — with API docs if available</i>
Payments	<i>Currencies, rails (card, UPI, bank, crypto), refunds, payouts, recurring</i>
Data to migrate	<i>From where, how much, what format</i>

5. Data and security

Sensitive data handled	<i>Identity documents, financial details, credentials, health — what and where stored</i>
Encryption requirements	<i>In transit · at rest · end-to-end (message stays private through servers)</i>
Compliance	<i>PCI DSS, DPDP, GDPR, KYC/AML, industry rules — what applies</i>
Audit and logging	<i>What must be traceable, for how long</i>
Hosting	<i>Your cloud account · ours · on-premises; region constraints</i>

6. Non-functional requirements

Scale	<i>Users, requests per second, data volume — today and in 12 months</i>
Availability	<i>Acceptable downtime; maintenance windows; disaster-recovery expectation</i>
Performance	<i>Page load, API latency, real-time requirements</i>
Languages and locales	<i>UI languages, currencies, time zones, RTL</i>
Accessibility	<i>WCAG level, if required</i>

7. Design

Existing brand or design system	<i>Logo, colours, fonts, component library, reference sites</i>
Design deliverables needed	<i>Wireframes · clickable prototype · full UI design · none (we have it)</i>

8. Edge cases what should happen when...

- | | |
|---|---|
| ☐ A payment succeeds but the webhook never arrives | ☐ Data is imported with duplicates or bad rows |
| ☐ The same request is sent twice | ☐ An admin needs to reverse an action |
| ☐ A third-party API is down for an hour | ☐ A customer requests deletion of their data |
| ☐ A user loses their device / second factor | ☐ Traffic spikes 10× for a day |

9. Testing, delivery and support

Acceptance criteria	<i>How you will decide it matches this document — list 3–5 checks</i>
Environments	<i>Dev · staging · production; who provides them</i>
Deliverables	<i>Source, repositories, CI, documentation, store listings, runbooks</i>
Ownership	<i>Code, cloud accounts, domains, app store accounts — who holds what</i>
Support window	<i>Defect window and support expectations after launch</i>
Confidentiality	<i>NDA required? Yes / no / already signed</i>

Prepared by · date

Approved by · date