

Roomio MVP Implementation Strategy & Roadmap

A Comprehensive 12-Week Plan to Launch

Document Information

Property	Value
Project Name	Roomio MVP
Version	1.0
Date	February 2026
Target	10-20 Small Hotels/Villas
Duration	12 Weeks

1. Executive Summary

Roomio is a comprehensive hotel management platform designed to streamline operations for small hotels and villas. This document outlines a concrete, implementation-oriented roadmap to take the product from concept to a launchable MVP within 12 weeks.

Project Goals

- **Primary Objective:** Build a fully functional MVP targeting 10-20 small hotels/villas
- **Tech Stack:** React Native (Expo) for Admin, Next.js/React PWA for Guest & Booking, Firebase ecosystem
- **Team Size:** 1 PM/Founder, 1 Product/UX Designer (part-time), 2-3 Engineers, 1 QA (part-time)
- **Launch Target:** Production-ready system with all core features operational

Key Deliverables Checklist

- Multi-tenant Admin Application
- Guest Web App (PWA)

- Public Booking Website
 - Basic AI Assistant (Text-based)
 - Operational Dashboards
 - Billing & Analytics
-

2. MVP Scope Definition

2.1 Target Users

User Type	Description
Property Owners	Manage multiple properties, view analytics
Managers	Oversee daily operations, staff management
Front Desk Staff	Handle check-in/check-out, reservations
Housekeeping	Manage room status and tasks
Guests	Access digital services, make requests

2.2 Must-Have Features (P0 - Critical)

The following features are essential for MVP launch:

Feature Category	Features
Multi-tenant Architecture	Organization-based data isolation
Admin App	Auth (owner, staff roles), Property setup, Room & inventory management
Reservation Management	Website + manual reservations, check-in/check-out flow
Guest Access	QR generation for Guest app, secure token-based access
Operations	Housekeeping tasks, Maintenance tickets, F&B orders
Billing	Manual payment recording, Invoices, Basic analytics
Guest Web App	Welcome screen, Digital menu, Property info, Request forms, Bill summary
Booking Page	Date selection, Availability check, Room selection, Guest details

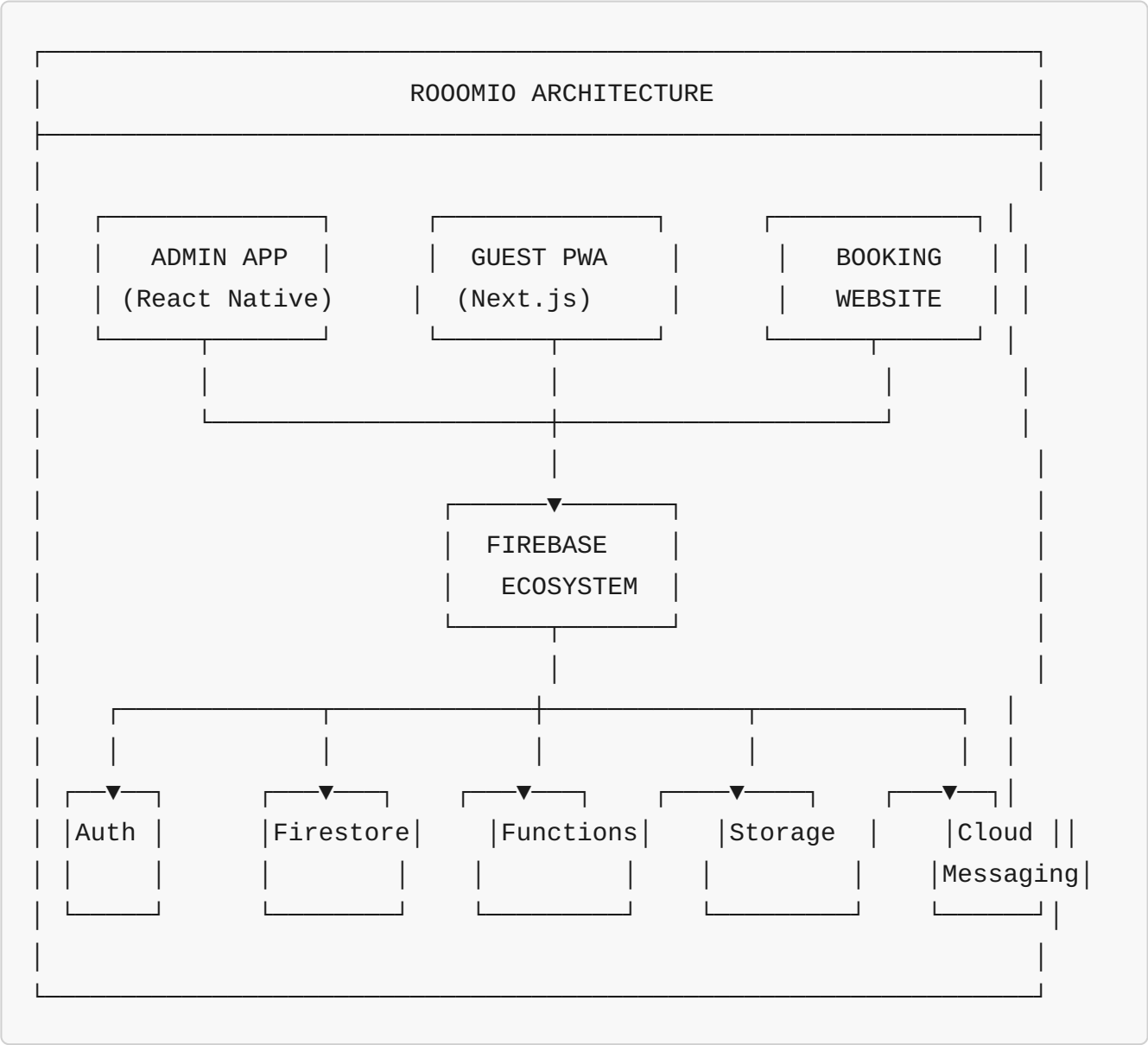
2.3 Nice-to-Have Features (P1 - Future)

These features can be deferred to v1.1 or v2.0:

Feature	Description
AI Assistant v1	Text-based chat with LLM + RAG
Voice Layer	Browser Speech-to-Text integration
Advanced Analytics	Detailed reporting and insights
Payment Gateway	Online payment processing
Multi-language	Extended language support

3. Technical Architecture & Stack

3.1 Technology Stack Overview



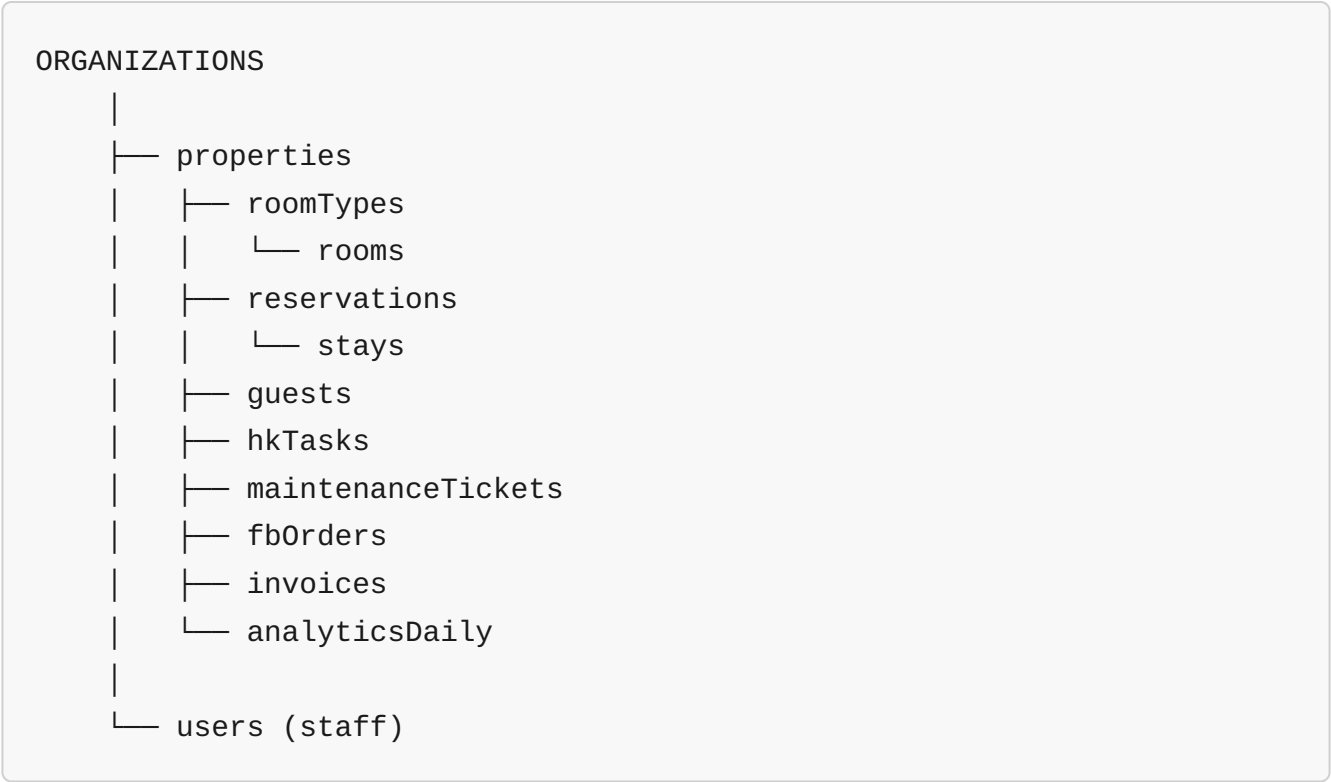
3.2 Frontend Technologies

Component	Technology	Purpose
Admin App	React Native (Expo)	Mobile application for staff
Guest App	Next.js PWA	Progressive web app for guests
Booking Website	Next.js/React	Public booking interface
Styling	Tailwind CSS	Modern responsive UI

3.3 Backend & Infrastructure

Component	Technology	Purpose
Database	Firestore	NoSQL document database
Authentication	Firebase Auth	User authentication
Serverless	Cloud Functions	Backend logic & APIs
Storage	Firebase Storage	File storage (logos, docs)
Hosting	Firebase Hosting	Web app hosting

3.4 Firestore Data Model



3.5 Key Architecture Decisions

Decision	Rationale
Multi-tenant Architecture	Enable multiple hotels to use the platform
Firebase Ecosystem	Rapid development, serverless, scalable
React Native (Expo)	Cross-platform mobile development
Next.js PWA	SEO-friendly, offline-capable guest app

Decision	Rationale
Firestore	Real-time sync, flexible schema

4. Team Composition & Governance

4.1 Team Structure

Role	Count	Responsibilities
PM/Founder	1	Product strategy, stakeholder management, prioritization
Product/UX Designer	1 (PT)	UI/UX design, user research, prototyping
Full-Stack Engineers	2-3	Frontend, backend, mobile development
QA Engineer	1 (PT)	Testing, bug tracking, quality assurance

4.2 Communication Cadence

Meeting	Frequency	Purpose
Daily Standup	Daily	Progress sync, blockers
Weekly Demo	Weekly	Feature showcase, feedback
Sprint Planning	Bi-weekly	Task assignment, estimates
Retrospective	Bi-weekly	Process improvement

5. 12-Week Implementation Plan

5.1 Phase Overview

Phase	Weeks	Focus Area	Key Outcomes
Phase 1	1-2	Foundation	Architecture, Data Model, Setup
Phase 2	3-8	Core Development	Admin, Booking, Guest App
Phase 3	9-10	AI & Features	AI Assistant, Multi-language
Phase 4	11-12	Launch Prep	QA, Hardening, Deployment

5.2 Detailed Week-by-Week Plan

Phase 1: Foundation (Weeks 1-2)

Week	Focus	Key Tasks	Deliverables
Week 1	Product Definition & Architecture	Finalize MVP scope, Draft PRD, Setup Firebase projects, Initialize repos	PRD Document, Project Setup
Week 2	Schema & Auth	Firestore schema, Security rules, Basic Admin shell, Seed data	Auth System, Admin Shell

Phase 2: Core Development (Weeks 3-8)

Week	Focus	Key Tasks	Deliverables
Week 3	Property Setup	Property config UI, Room types, Room inventory	Property Module
Week 4	Booking System	Rate plans, Tax config, Booking page, Reservation management	Booking System
Week 5	Check-in/out	Stay model, Check-in flow, QR generation, Check-out flow	Check-in/out System
Week 6	Guest App	Guest PWA, Property info, Digital menu, Requests	Guest Web App
Week 7	Operations	Housekeeping board, Maintenance panel, F&B orders	Operations Dashboard
Week 8	Billing & Analytics	Billing model, Invoices, Analytics dashboard	Billing & Analytics

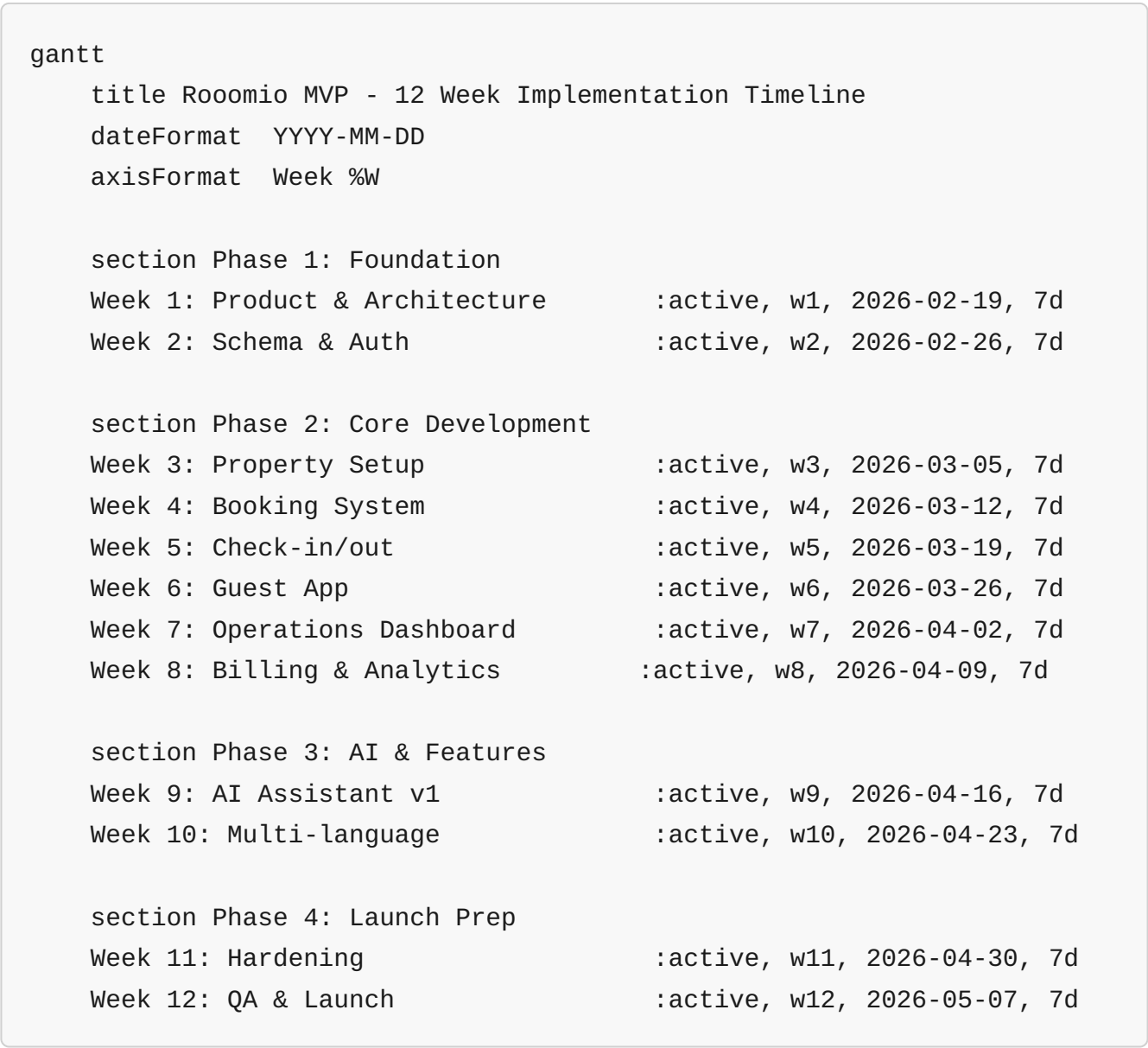
Phase 3: AI & Features (Weeks 9-10)

Week	Focus	Key Tasks	Deliverables
Week 9	AI Assistant	LLM integration, Intent mapping, Chat UI	AI Chat v1
Week 10	Multi-language	i18n setup, EN/HI translations, Voice input	Multi-language Support

Phase 4: Launch Prep (Weeks 11-12)

Week	Focus	Key Tasks	Deliverables
Week 11	Hardening	Security review, Validation, Performance optimization	Secure & Optimized System
Week 12	QA & Launch	E2E testing, Pilot onboarding, Launch checklist	Production Launch

6. Gantt Chart: 12-Week Timeline



7. Feature Priority Matrix

7.1 Admin Features

Feature	Priority	User Impact	Effort	Status
Multi-tenant Architecture	P0	Critical	High	To Do
Property Setup & Config	P0	Critical	Medium	To Do
Room & Inventory Management	P0	Critical	Medium	To Do
Reservation Management	P0	Critical	High	To Do
Check-in/Check-out Flow	P0	Critical	High	To Do
QR Generation	P0	High	Low	To Do
Housekeeping Tasks	P0	High	Medium	To Do
Maintenance Tickets	P0	High	Medium	To Do
F&B Orders	P0	High	Medium	To Do
Billing & Payments	P0	High	Medium	To Do
Basic Analytics	P0	Medium	Medium	To Do

7.2 Guest Features

Feature	Priority	User Impact	Effort	Status
Welcome & Stay Info	P0	Critical	Low	To Do
Digital Menu	P0	High	Medium	To Do
Property Info (WiFi, Policies)	P0	High	Low	To Do
Request Forms	P0	High	Medium	To Do
Bill Summary View	P0	High	Medium	To Do
Multi-language (EN+HI)	P0	High	Medium	To Do
AI Assistant (Text)	P1	Medium	High	To Do
Voice Input	P1	Medium	Medium	To Do

7.3 Booking Features

Feature	Priority	User Impact	Effort	Status
Date Selection	P0	Critical	Medium	To Do
Availability Check	P0	Critical	High	To Do
Room Type Selection	P0	Critical	Medium	To Do
Guest Details Form	P0	Critical	Low	To Do
Reservation Creation	P0	Critical	Medium	To Do

8. Milestones & Key Dates

```

timeline
    title Roomio MVP - Key Milestones
    2026-02-19 : Project Kickoff
    2026-02-26 : Architecture Complete
                  : Auth & Schema Ready
    2026-03-05 : Property Module Live
    2026-03-19 : Booking System Live
                  : Check-in/out Ready
    2026-04-02 : Guest App Launched
                  : Operations Dashboard Live
    2026-04-16 : Billing & Analytics Ready
    2026-04-23 : AI Assistant v1 Live
    2026-04-30 : Multi-language Support
    2026-05-07 : MVP Launch

```

9. Risk Register

Risk Item	Probability	Impact	Mitigation Strategy
Scope Creep	High	High	Strict priority gating, weekly scope reviews
Resource Availability	Medium	High	Part-time resources confirmed, backup options identified

Risk Item	Probability	Impact	Mitigation Strategy
Technical Complexity	Medium	Medium	Early architecture proofs, Firebase expertise leveraged
Integration Issues	Low	Medium	Phased integration testing, mock services
Timeline Delays	Medium	High	Buffer in Phase 4, daily standups for early detection

10. Success Metrics

10.1 Technical Metrics

Metric	Target
System Uptime	99.5%
Page Load Time	< 3 seconds
API Response Time	< 500ms
Test Coverage	> 80%

10.2 Business Metrics

Metric	Target
Pilot Properties	10-20
User Satisfaction	> 4.0/5.0
Feature Adoption	> 70%
System Stability	< 2 critical bugs

11. Next Steps

1. **Confirm Team Availability** - Finalize resource allocation
2. **Set Up Development Environment** - Initialize Firebase projects and repositories
3. **Begin Week 1 Activities** - Start with product definition and architecture
4. **Schedule Weekly Reviews** - Establish communication cadence

