

Nomado Travel Platform - Development Gantt Chart

Target: Complete Travel Platform from Current State to Production



Week 1: Backend Infrastructure & API Completion

Days 1-2: Environment & Configuration

- ☐ Create .env template and documentation
- ☐ Set up proper logging system
- ☐ Configure environment-specific settings
- ☐ Add database migrations system

Days 3-4: API Enhancement

- ☐ Complete missing API endpoints (hotels, reviews, search)
- ☐ Add pagination to all list endpoints
- ☐ Implement proper error handling and validation
- ☐ Add API documentation with Swagger

Days 5-7: Authentication & Security

- ☐ Enhance JWT implementation with refresh tokens
- ☐ Add role-based access control (admin, user)
- ☐ Implement password reset functionality
- ☐ Add rate limiting and request validation

Week 2: Database & Business Logic

Days 1-3: Database Enhancement

- ☐ Complete all table relationships and constraints
- ☐ Add indexes for performance optimization
- ☐ Implement database backup/restore procedures
- ☐ Add data seeding for development/testing

Days 4-5: Business Logic Implementation

- ☐ Hotel/accommodation booking system
- ☐ Payment integration planning
- ☐ Booking cancellation and modification
- ☐ User profile management

Days 6-7: Testing Infrastructure

- ☐ Set up unit testing framework
- ☐ Write tests for repositories and services
- ☐ Set up integration testing
- ☐ Add test database setup

Week 3: Advanced Features

Days 1-3: Search & Filtering

- ☐ Implement advanced search functionality
- ☐ Add filtering by price, rating, location
- ☐ Implement geo-location based search
- ☐ Add sorting options

Days 4-5: File Management

- ☐ Image upload functionality
- ☐ File storage solution (local/cloud)
- ☐ Image optimization and resizing
- ☐ Document upload for visa services

Days 6-7: Notifications & Communication

- ☐ Email service integration
- ☐ SMS notification system
- ☐ In-app notification system
- ☐ Email templates for bookings

Week 4: API Optimization & Documentation

Days 1-3: Performance & Optimization

- ☐ Database query optimization
- ☐ API response caching
- ☐ Connection pooling
- ☐ Load testing and performance tuning

Days 4-7: Documentation & Developer Experience

- ☐ Complete API documentation
- ☐ Add code comments and documentation
- ☐ Create developer setup guide
- ☐ API versioning implementation

Phase 2: Modern Frontend Development (Weeks 5-8)

Week 5: Frontend Architecture & Setup

Days 1-2: Technology Stack Decision

- ☐ Choose framework (React/Vue.js/Angular)
- ☐ Set up build tools (Vite/Webpack)
- ☐ Configure TypeScript
- ☐ Set up UI component library (Tailwind CSS/Material-UI)

Days 3-4: Project Structure

- ☐ Create proper folder structure
- ☐ Set up routing system
- ☐ Configure state management (Redux/Zustand/Pinia)
- ☐ Set up HTTP client (Axios/Fetch API)

Days 5-7: Core Components

- ☐ Navigation component
- ☐ Layout components
- ☐ Common UI components (buttons, forms, modals)
- ☐ Loading and error states

Week 6: Authentication & User Management

Days 1-3: Auth System Frontend

- ☐ Login/Register forms with validation
- ☐ JWT token management
- ☐ Protected routes implementation
- ☐ User session handling

Days 4-5: User Dashboard

- ☐ User profile page
- ☐ Booking history
- ☐ Profile editing functionality
- ☐ Account settings

Days 6-7: Admin Panel (Basic)

- ☐ Admin authentication
- ☐ User management interface
- ☐ Basic analytics dashboard
- ☐ Content management system

Week 7: Core Features Frontend

Days 1-3: Destination & Service Pages

- ☐ Destination listing page with filters
- ☐ Individual destination detail pages
- ☐ Service catalog with categories
- ☐ Image galleries and carousels

Days 4-5: Search & Discovery

- ☐ Advanced search interface
- ☐ Filter sidebar
- ☐ Search results page
- ☐ Map integration (Google Maps/Mapbox)

Days 6-7: Booking System Frontend

- ☐ Booking flow (multi-step form)
- ☐ Date picker integration
- ☐ Price calculation
- ☐ Booking confirmation

Week 8: Advanced Frontend Features

Days 1-3: Interactive Features

- ☐ Reviews and ratings system
- ☐ Wishlist/favorites functionality
- ☐ Social sharing
- ☐ Live chat integration

Days 4-5: Mobile Responsiveness

- ☐ Mobile-first responsive design
- ☐ Touch gestures and interactions
- ☐ Mobile navigation menu
- ☐ Progressive Web App (PWA) setup

Days 6-7: Performance Optimization

- ☐ Code splitting and lazy loading
- ☐ Image optimization
- ☐ Bundle size optimization
- ☐ SEO optimization

Phase 3: Integration & Advanced Features (Weeks 9-10)

Week 9: Third-party Integrations

Days 1-3: Payment Integration

- ☐ Stripe/PayPal integration
- ☐ Payment form frontend
- ☐ Webhook handling
- ☐ Payment confirmation flow

Days 4-5: External APIs

- ☐ Weather API integration
- ☐ Currency conversion API
- ☐ Flight/hotel booking APIs
- ☐ Visa requirement APIs

Days 6-7: Communication Features

- ☐ Email notification system
- ☐ SMS integration
- ☐ Push notifications
- ☐ Customer support chat

Week 10: Content Management & Admin Features

Days 1-3: Admin Panel Enhancement

- ☐ Complete admin dashboard
- ☐ Content management (destinations, hotels)
- ☐ User management and moderation
- ☐ Analytics and reporting

Days 4-5: Content Features

- ☐ Blog/travel guide system
- ☐ Image management system
- ☐ SEO-friendly URLs
- ☐ Content scheduling

Days 6-7: Business Intelligence

- ☐ Booking analytics
- ☐ Revenue reporting
- ☐ User behavior tracking
- ☐ Performance metrics

Phase 4: Testing, Deployment & Production (Weeks 11-12)

Week 11: Comprehensive Testing

Days 1-2: Backend Testing

- ☐ Complete unit test coverage
- ☐ Integration testing
- ☐ API endpoint testing
- ☐ Database testing

Days 3-4: Frontend Testing

- ☐ Component testing
- ☐ End-to-end testing (Cypress/Playwright)
- ☐ User journey testing
- ☐ Cross-browser testing

Days 5-7: Security & Performance Testing

- ☐ Security vulnerability scanning
- ☐ Load testing and stress testing
- ☐ Mobile device testing
- ☐ Accessibility testing

Week 12: Production Deployment

Days 1-2: Production Infrastructure

- ☐ Set up production servers (AWS/GCP/Azure)
- ☐ Configure production database
- ☐ Set up CDN for static assets
- ☐ Configure SSL certificates

Days 3-4: Deployment Pipeline

- ☐ Set up CI/CD pipeline
- ☐ Automated testing in pipeline
- ☐ Environment configuration
- ☐ Database migration scripts

Days 5-6: Go-Live Preparation

- ☐ Final testing in production environment
- ☐ DNS configuration
- ☐ Monitoring and logging setup
- ☐ Backup and disaster recovery

Day 7: Launch & Post-Launch

- ☐ Production deployment

- ☐ Launch announcement
- ☐ Monitor system performance
- ☐ User feedback collection

Resource Allocation & Team Structure

Recommended Team Size: 2-3 Developers

- **Backend Developer** (1 person): Focus on API, database, integrations
- **Frontend Developer** (1 person): Focus on UI/UX, user experience
- **Full-stack Developer** (1 person): Support both areas, DevOps, testing

Technology Stack Recommendations

Backend (Current + Enhancements)

- **Language:** Go (current)
- **Framework:** Gorilla Mux (current) + Gin (optional upgrade)
- **Database:** PostgreSQL (current)
- **Cache:** Redis
- **Search:** Elasticsearch (optional)
- **File Storage:** AWS S3 or local with backup

Frontend (New)

- **Framework:** React.js with TypeScript
- **Build Tool:** Vite
- **UI Library:** Tailwind CSS + Headless UI
- **State Management:** Zustand or React Query
- **Maps:** Mapbox GL JS
- **Testing:** Vitest + Cypress

DevOps & Infrastructure

- **Containerization:** Docker (current)
- **Orchestration:** Docker Compose (current)
→ Kubernetes (future)
- **Cloud Provider:** AWS/GCP/Azure
- **CI/CD:** GitHub Actions or GitLab CI
- **Monitoring:** Prometheus + Grafana
- **Logging:** ELK Stack or Loki

Key Milestones & Deliverables

Week 4: Complete backend API with documentation

Week 6: User authentication and basic frontend

Week 8: Complete user-facing features

Week 10: Admin panel and business features

Week 12: Production-ready application

Risk Mitigation

High-Risk Items

Payment integration complexity

Third-party API dependencies

Performance under load

Security vulnerabilities

Mitigation Strategies

- Start payment integration early
- Have fallback options for external APIs
- Implement caching and optimization early
- Regular security audits and testing

Success Metrics

Technical Metrics

API response time < 200ms

Frontend page load time < 3 seconds

99.9% uptime

Zero critical security vulnerabilities

Business Metrics

User registration conversion > 15%

Booking completion rate > 70%

Customer satisfaction score > 4.5/5

Support ticket resolution < 24 hours

Next Steps to Get Started

- 1. **Week 1 Day 1:** Create project roadmap and assign tasks
- 2. **Set up development environment:** Ensure all team members can run the project locally
- 3. **Choose frontend technology:** Make final decision on React/Vue/Angular
- 4. **Create project backlog:** Break down tasks into smaller, manageable tickets
- 5. **Set up project management:** Use tools like Jira, Trello, or GitHub Projects

This Gantt chart provides a structured path from your current state to a production-ready travel platform. Adjust timelines based on your team size and complexity requirements.