Glamping Website Development Guide

Complete Technical Roadmap for Building a Modern Booking Platform

---

🎯 Project Overview

Goal: Create a professional glamping website with location information, pricing, descriptions, image galleries, and integrated reservation system.

Target Timeline: 8-12 weeks for full MVP **Estimated Budget:** $15,000-$30,000 (if outsourcing development)

🛠 Recommended Technology Stack

Frontend Technologies

- **Framework:** React.js with Next.js 14+
- **Styling:** Tailwind CSS + Headless UI
- **Language:** TypeScript
- **State Management:** Zustand or React Query
- **Forms:** React Hook Form + Zod validation

Why this stack?

- Server-side rendering for excellent SEO
- Built-in optimization features
- Large ecosystem for booking integrations
- Excellent developer experience

Backend Technologies

- **Runtime:** Node.js 18+
- **Framework:** Express.js or Fastify
- **Language:** TypeScript
- **ORM:** Prisma or TypeORM
- **Authentication:** JWT + refresh tokens
- **File Upload:** Multer + Cloudinary

Database & Infrastructure

- **Primary Database:** PostgreSQL 14+
- **Caching:** Redis (for sessions & booking locks)
- **File Storage:** Cloudinary or AWS S3
- **Email Service:** SendGrid or Resend
- **Payment Processing:** Stripe
- **Hosting:** Vercel (frontend) + Railway (backend)

📋 Development Phases

Phase 1: Foundation & Setup (2-3 weeks)

Week 1: Project Initialization

□ Set up Next.js project with TypeScript
□ Configure ESLint, Prettier, and Husky
□ Set up PostgreSQL database
□ Create initial database schema
□ Set up version control (Git + GitHub)
□ Configure development environment

Week 2: Backend Foundation

□ Create Express.js API server
□ Set up database ORM (Prisma recommended)
□ Implement user authentication system
□ Create basic CRUD endpoints
□ Set up error handling middleware
□ Configure environment variables

Core Database Schema

```sql
Users Table:
- id, email, password_hash, first_name, last_name
- phone, created_at, updated_at, role (guest/admin)

Properties Table:
- id, name, description, max_guests, base_price
- amenities, images, location_data, created_at

Bookings Table:
- id, user_id, property_id, check_in, check_out
- total_price, status, guest_count, created_at

Availability Table:
- id, property_id, date, available, custom_price
```

Phase 2: Core Features Development (3-4 weeks)

Week 3: Property Management

- □ Create property listing pages
- □ Implement image gallery component
- □ Build property detail views
- □ Add property search and filtering
- □ Create responsive design system

Week 4: Booking System Foundation

- □ Build availability calendar component
- □ Implement date range selection
- □ Create booking validation logic
- □ Add pricing calculation engine
- □ Build booking confirmation flow

Week 5: User Management

- □ Implement user registration/login
- □ Create user dashboard
- □ Add booking history view
- □ Build profile management
- □ Implement password reset functionality

Week 6: Admin Panel

- □ Create admin dashboard
- □ Build property management interface
- □ Add booking management system
- □ Implement availability calendar management
- □ Create reporting and analytics views

Phase 3: Advanced Features (2-3 weeks)

Week 7: Payment Integration

- □ Set up Stripe payment processing
- □ Implement secure checkout flow
- □ Add payment confirmation handling
- □ Create refund management system
- □ Build invoice generation

Week 8: Communication System

- □ Set up email service (SendGrid)
- □ Create booking confirmation emails
- □ Add reminder email automation
- □ Build contact form
- □ Implement SMS notifications (optional)

Week 9: Enhanced Features

- □ Add seasonal pricing rules
- □ Implement discount codes system
- □ Create review and rating system
- □ Add calendar sync (iCal/Google)
- □ Build advanced search filters

Phase 4: Optimization & Launch (1-2 weeks)

Week 10: Performance & SEO

- □ Optimize images and loading times
- □ Implement proper SEO meta tags
- □ Add structured data markup
- □ Configure sitemap generation
- □ Set up Google Analytics

Week 11: Security & Testing

- □ Conduct security audit
- □ Implement rate limiting
- □ Add comprehensive error logging
- □ Perform load testing
- □ Create backup procedures

Week 12: Deployment & Monitoring

- □ Set up production environment
- □ Configure domain and SSL
- □ Implement monitoring and alerts
- □ Create documentation
- □ Launch and monitor

🏗️ Technical Architecture

Frontend Structure

```
src/
├── components/
│   ├── ui/          # Reusable UI components
│   ├── forms/       # Form components
│   ├── booking/     # Booking-specific components
│   └── admin/       # Admin panel components
├── pages/
│   ├── api/         # Next.js API routes
│   ├── properties/  # Property pages
│   ├── booking/     # Booking flow
│   └── admin/       # Admin pages
├── hooks/           # Custom React hooks
├── utils/           # Utility functions
└── styles/          # Global styles
```

Backend Structure

```
src/
├── routes/
│   ├── auth.js       # Authentication endpoints
│   ├── properties.js # Property CRUD
│   ├── bookings.js   # Booking management
│   └── users.js      # User management
├── middleware/       # Authentication, validation
├── models/           # Database models
├── services/         # Business logic
└── utils/            # Helper functions
```

🔐 Security Considerations

Essential Security Measures

☐ Input validation on all endpoints
☐ SQL injection prevention (use ORM)
☐ XSS protection with proper sanitization
☐ CSRF tokens for form submissions
☐ Rate limiting on API endpoints
☐ Secure session management
☐ HTTPS enforcement
☐ Environment variable protection

Payment Security

☐ Never store credit card information
☐ Use Stripe's secure tokenization
☐ Implement PCI compliance measures
☐ Add fraud detection rules
☐ Secure webhook handling

📱 Key Features Checklist

Public Features

- [ ] Property listings with detailed descriptions
- [ ] High-quality image galleries
- [ ] Interactive availability calendar
- [ ] Real-time pricing calculator
- [ ] Mobile-responsive design
- [ ] Fast loading times (< 3 seconds)
- [ ] SEO-optimized pages
- [ ] Contact forms and information

Booking System

- [ ] Date range selection
- [ ] Guest count specification
- [ ] Real-time availability checking
- [ ] Pricing breakdown display
- [ ] Secure payment processing
- [ ] Booking confirmation emails
- [ ] Cancellation management
- [ ] Booking modifications

Admin Panel

- [ ] Property management interface
- [ ] Booking calendar overview
- [ ] Guest communication tools
- [ ] Revenue reporting
- [ ] Availability management
- [ ] Pricing rule configuration
- [ ] User management
- [ ] System analytics

💰 Cost Estimation

Development Costs (if outsourcing)

- **Frontend Development:** $8,000 - $15,000
- **Backend Development:** $6,000 - $12,000
- **Integration & Testing:** $2,000 - $4,000
- **Design & UX:** $3,000 - $6,000

Monthly Operating Costs

- **Hosting (Vercel + Railway):** $20 - $100
- **Database (PostgreSQL):** $15 - $50
- **CDN & Storage (Cloudinary):** $10 - $30
- **Email Service (SendGrid):** $15 - $50
- **Payment Processing (Stripe):** 2.9% + $0.30 per transaction
- **Domain & SSL:** $15 - $30 annually

🚀 Deployment Strategy

Staging Environment

□ Set up staging server for testing
□ Configure CI/CD pipeline
□ Implement automated testing
□ Create deployment scripts

Production Environment

□ Configure production database
□ Set up monitoring and logging
□ Implement backup procedures
□ Configure security settings

Launch Checklist

□ Domain setup and SSL certificates
□ Google Analytics and Search Console
□ Social media integration
□ Email marketing setup
□ Customer support system
□ Legal pages (terms, privacy policy)

📚 Additional Resources

Learning Resources

- Next.js Documentation: https://nextjs.org/docs
- React Documentation: https://react.dev
- Stripe Integration Guide: https://stripe.com/docs
- PostgreSQL Tutorial: https://www.postgresql.org/docs/

Useful Tools

- **Design:** Figma for mockups
- **Database:** TablePlus or pgAdmin
- **API Testing:** Postman or Insomnia
- **Monitoring:** Sentry for error tracking
- **Performance:** Google PageSpeed Insights

⚡ Quick Start Commands

Initial Setup

```bash
# Frontend (Next.js)
npx create-next-app@latest glamping-site --typescript --tailwind

# Backend (Express.js)
mkdir glamping-api && cd glamping-api
npm init -y
npm install express prisma typescript ts-node

# Database Setup
npx prisma init
npx prisma migrate dev --name init
```

Development Commands

```bash
# Start frontend development server
npm run dev

# Start backend development server
npm run dev:server

# Database operations
npx prisma studio       # Database GUI
npx prisma migrate dev  # Apply migrations
```

This guide serves as your complete roadmap for developing a professional glamping booking website. Adjust timelines and features based on your specific requirements and resources.