# Build a SaaS in 48 Hours

Hour-by-Hour Blueprint • Tech Stack • Launch Checklist

**END OF CODING**
Build Apps with AI • No Code Experience Needed

# Your 48-Hour SaaS Journey

This blueprint has been tested on dozens of successful SaaS launches. By combining AI tools like Bolt.new, Cursor, and modern platforms like Supabase and Vercel, you can go from idea to paying customers in a single weekend. Follow this hour-by-hour schedule and ship your product fast.

## Recommended Tech Stack

**Frontend:** React + TypeScript + Tailwind CSS + shadcn/ui

**AI Builder:** Bolt.new (rapid prototyping) + Cursor (refinement)

**Backend:** Supabase (Database + Auth + Storage + Edge Functions)

**Payments:** Stripe (subscriptions + checkout)

**Deployment:** Vercel (frontend) + Supabase (backend)

**Analytics:** Vercel Analytics + PostHog

**Email:** Resend or SendGrid

# Hours 0-8: Planning & Setup

## Hour 0-2: Idea Validation

- ☐ Define your core value proposition in one sentence
- ☐ Identify your target customer and their pain point
- ☐ Research 3-5 competitors - what will you do differently?
- ☐ Sketch a simple 3-page flow: Landing → Sign up → Core feature
- ☐ List your MVP features (max 3 core features only)

## Hour 2-4: Technical Planning

- ☐ Create accounts: Supabase, Vercel, Stripe (test mode)
- ☐ Set up Bolt.new project with your tech stack
- ☐ Design database schema (users, subscriptions, core data models)
- ☐ Plan API endpoints you'll need
- ☐ Choose a domain name and register it (optional but recommended)

## Hour 4-6: Project Scaffolding

- ☐ Generate project structure in Bolt.new with detailed prompt
- ☐ Set up Supabase database tables and RLS policies
- ☐ Configure authentication (email + Google OAuth)

- [ ] Create .env file with all API keys
- [ ] Initialize Git repo and first commit

## Hour 6-8: Core Infrastructure

- [ ] Implement authentication flow (login, signup, logout)
- [ ] Create protected routes and middleware
- [ ] Set up error handling and loading states
- [ ] Configure Supabase client for API calls
- [ ] Test auth flow end-to-end

# Hours 8-24: Core Features

## Hour 8-12: Main Feature #1

- [ ] Build the UI for your primary feature using AI
- [ ] Connect to database with CRUD operations
- [ ] Add form validation and error handling
- [ ] Implement real-time updates if applicable (Supabase realtime)
- [ ] Test all happy paths and edge cases

## Hour 12-16: Main Feature #2

- [ ] Build second core feature
- [ ] Ensure data relationships work correctly
- [ ] Add user permissions (users can only see their own data)
- [ ] Implement search/filtering if needed
- [ ] Test with multiple user accounts

## Hour 16-20: Main Feature #3 & Polish

- [ ] Complete third feature (or enhance first two)
- [ ] Add dashboard/overview page
- [ ] Implement user settings page

- [ ] Create onboarding flow for new users
- [ ] Add helpful empty states and tooltips

## Hour 20-24: User Experience

- [ ] Improve UI consistency (spacing, colors, typography)
- [ ] Add loading skeletons for better perceived performance
- [ ] Implement toast notifications for user actions
- [ ] Make responsive for mobile
- [ ] Run through entire user journey and fix issues

# Hours 24-40: Polish & Payments

## Hour 24-28: Stripe Integration

- [ ] Set up Stripe products and pricing
- [ ] Implement Stripe Checkout (use pre-built checkout page)
- [ ] Create webhook endpoint for subscription events
- [ ] Update user permissions based on subscription status
- [ ] Test payment flow with Stripe test cards

## Hour 28-32: Landing Page

- [ ] Create compelling hero section with clear value prop
- [ ] Add features section (benefits, not just features)
- [ ] Include social proof (if you have any early users)
- [ ] Add pricing section with clear CTAs
- [ ] Create FAQ section addressing common objections

## Hour 32-36: Content & Legal

- [ ] Write Terms of Service (use template + customize)
- [ ] Create Privacy Policy (GDPR compliant)
- [ ] Add About page explaining your mission

- [ ] Write help documentation for key features
- [ ] Set up customer support email/system

## Hour 36-40: Final Polish

- [ ] Fix all console errors and warnings
- [ ] Optimize images and bundle size
- [ ] Add meta tags for SEO (title, description, OG images)
- [ ] Set up Google Analytics or PostHog
- [ ] Test on different browsers and devices

# Hours 40-48: Launch

## Hour 40-44: Deployment

- ☐ Deploy to Vercel (connect GitHub repo for auto-deploy)
- ☐ Configure custom domain and SSL
- ☐ Set production environment variables
- ☐ Enable Vercel Analytics and logging
- ☐ Run final production tests

## Hour 44-46: Pre-Launch Checklist

- ☐ Test complete user journey one more time
- ☐ Verify all payment flows work in production
- ☐ Check all external links and email templates
- ☐ Set up error monitoring (Sentry or Vercel)
- ☐ Prepare launch tweet/post with screenshot

## Hour 46-48: Launch & Promote

- ☐ Post on Twitter/X with demo video
- ☐ Submit to Product Hunt (schedule for 12:01am PT)
- ☐ Share in relevant Reddit communities (provide value, don't spam)

- [ ] Post in Indie Hackers, HackerNews (Show HN)
- [ ] Email your network and ask for feedback
- [ ] Join relevant Slack/Discord communities and share (where allowed)

## Keys to Success

• **Cut scope ruthlessly:** You can't build everything in 48 hours. Three core features that work perfectly beat ten half-finished features.

• **Use AI for speed:** Let Bolt.new generate boilerplate, use Cursor for refinements. Don't write repetitive code manually.

• **Leverage templates:** Use shadcn/ui components, Stripe's pre-built checkout, Supabase's auth UI components.

• **Don't get stuck:** If something takes more than 2 hours, find a simpler solution or skip it for v1.

• **Ship imperfect:** You can iterate after launch. Done is better than perfect for v1.

• **Get feedback fast:** Launch to a small audience first, learn, then expand promotion.