Prompt Machine - MVP-First Development Roadmap

Overview

Build Prompt Machine incrementally, starting with a revenue-generating MVP in 2 weeks, then adding features based on user feedback and revenue.

Version Release Schedule

o Version 1.0.0 - MVP (Target: 2 weeks)

Goal: Get a working product that can generate revenue immediately.

Core Features:

- 1. User authentication (login/logout)
- 2. Project management (create/list)
- 3. Claude chat integration
- 4. Basic prompt builder
- 5. Simple HTML deployment
- 6. Google AdSense integration

Revenue Potential: \$100-500/month from ads



Version 1.1.0 - First Enhancement (Target: 1 week after 1.0)

Goal: Improve user experience based on initial feedback.

New Features:

- 1. Email notifications (SendGrid)
- 2. Tailwind CSS styling
- 3. Basic prompt templates
- 4. Improved error handling
- 5. Simple analytics (page views)

Revenue Potential: \$500-1000/month (better retention)

Version 1.2.0 - User Requested Features (Target: 2 weeks after 1.1)

Goal: Add the top 5 features users actually ask for.

Planned Features:

- 1. User dashboard
- 2. Prompt versioning
- 3. Basic A/B testing
- 4. SSL for deployed tools
- 5. Stripe payment integration

Revenue Potential: \$1000-2000/month (paid tiers)

✓ Version 2.0.0 - Scale & Optimize (Target: 1 month after 1.2)

Goal: Handle growth and reduce costs.

Major Upgrades:

- 1. Redis caching layer
- 2. Multi-LLM support (Claude, GPT, Llama)
- 3. Job queue for deployments
- 4. Advanced analytics
- 5. File export (PDF, DOCX)
- 6. AWS cost monitoring
- 7. Two-factor authentication

Revenue Potential: \$5000-10000/month

■ Version 3.0.0 - Enterprise Features (Target: 2 months after 2.0)

Goal: Serve larger customers and teams.

Enterprise Features:

- 1. Custom domains
- 2. Team collaboration
- 3. API access

- 4. Infrastructure Al
- 5. Advanced monetization
- 6. White-label options
- 7. SLA guarantees

Revenue Potential: \$20000+/month

Development Principles

1. Revenue First

Every version must improve revenue potential. If a feature doesn't help make money or retain users, it waits.

2. User Feedback Driven

After 1.0.0, let users tell you what to build next. Don't guess.

3. Technical Debt is OK

In early versions, choose speed over perfection. Refactor when revenue justifies it.

4. Claude Code Maximum Usage

For each version, we'll specify exactly what Claude Code should build to maximize development speed.

5. Launch Fast, Iterate Faster

Each version should take no more than stated time. Cut features to hit deadlines.

Success Metrics

Version 1.0.0

- ✓ 10 active users
- ✓ 1 paying customer
- ✓ \$100 in revenue

Version 1.1.0

- ✓ 50 active users
- ✓ 5 paying customers
- ✓ \$500 in revenue

Version 1.2.0

- ✓ 200 active users
- ✓ 20 paying customers
- ✓ \$1000 in revenue

Version 2.0.0

- ✓ 1000 active users
- ✓ 100 paying customers
- ✓ \$5000 in revenue

Next Steps

- 1. Start with Version 1.0.0 MVP Guide (Artifact 02-NEW)
- 2. Use Claude Code for maximum speed
- 3. Launch in 2 weeks
- 4. Get user feedback
- 5. Build Version 1.1.0

Remember: Perfect is the enemy of done. Ship fast, learn faster!