Pitch Perfect

*Building Hackathon-Winning
Pitch Decks with AI*

Cursor **Marp** **AI-Powered**

The Hackathon Presentation Problem

Common Challenges

- **Developers prioritize building over pitching**

- **Limited design skills** in technical teams

- **Last-minute presentation creation** leads to poor quality

- **Lack of visual storytelling** undermines great projects

- **Time spent on slides = less time coding**

Hackathon Pitch Challenges

Time Constraints | Design Limitations | Prioritize Code over Slides | Last-Minute Rush

Your Guides for Today

Dr. Marton Csernai

- **PhD in Computer Science**
- 15+ years development experience
- Former Research Lead at KILT Protocol
- Currently building Gigentic "Escrow-as-a-Service"
- Superteam Germany member

Alexander Schmitt

- **Serial entrepreneur with successful exit**
- Raised €3.5M for previous venture
- Deep expertise in blockchain
- Co-founder of Gigentic
- Experienced in hackathon presentations

Real-World Application: We're using this exact workflow to create our Breakout Hackathon pitch deck

Built for Builders: Designed for developers who want professional results without sacrificing coding time

Our Solution: Cursor + Marp Workflow

Project Docs → Cursor AI → Draft Content → Marp Markdown → Visual Slides → Export Formats

The Power of Cursor AI

- **Intelligent content generation** from project materials
- **Code-centric environment** familiar to developers
- **Iterative refinement** with targeted prompts
- **Collaborative workflow** for team input

The Magic of Marp

- **Markdown-to-presentation** conversion
- **Version control friendly** - works with Git
- **Customizable themes** for brand consistency
- **Mermaid diagram integration** for architecture visuals
- **Export to multiple formats** (PDF, PPTX, HTML)

Key Benefit: Create professional presentations while staying in your development environment

What is Marp?

Markdown Presentation Ecosystem

- **Marp = Markdown Presentation Ecosystem**
- Write slides in Markdown, render as presentations
- VS Code extension with live preview
- CLI for automation and CI/CD integration
- Built for developers who prefer text over GUI

How Marp Works

```
---
marp: true
theme: default
---

# Slide 1 Title

Content for first slide


---

# Slide 2 Title

- Bullet points
- More content
```
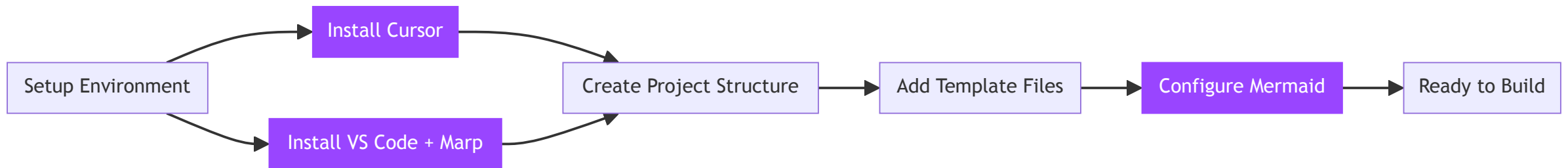
- **Front matter** for configuration
- **Triple dash (---)** separates slides
- **Standard Markdown** for content formatting

Developer-First: Marp treats presentations as code - write, version, collaborate, and deploy
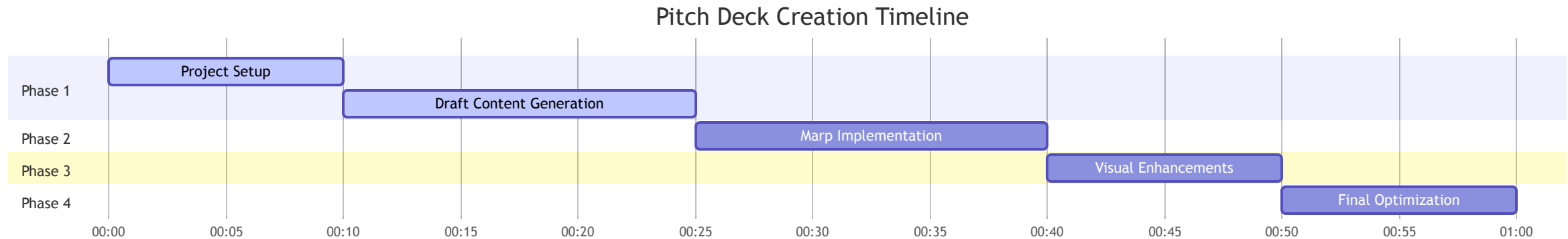
Technical Implementation


```
Setup Environment → Install Cursor → Create Project Structure
Setup Environment → Install VS Code + Marp → Create Project Structure
Create Project Structure → Add Template Files → Configure Mermaid → Ready to Build
```

Full Environment: Complete setup in under 15 minutes, even for newcomers

Workflow: From Project to Pitch

Pitch Deck Creation Timeline

Phase 1	Project Setup, Draft Content Generation	
Phase 2	Marp Implementation	
Phase 3	Visual Enhancements	
Phase 4	Final Optimization	

Timeline: 00:00 – 01:00

AI Prompt Strategy

- Start with **project description** & PRD
- Generate **slide structure** before details
- Use **targeted prompts** for specific sections
- Leverage AI for **visual suggestions**

Marp Implementation Tips

- Maintain **consistent brand styling**
- Utilize **Mermaid for technical diagrams**
- Create **reusable component styles**
- Add **responsive layout** considerations
- **Minimize text** / Maximize impact

Advantages of Our Approach

For Developers

- **Stay in your coding environment**
- **Version control** your presentation
- **Collaborate** using Git workflows
- **Automate repetitive tasks** with AI
- **Focus on code** while AI handles content

For Teams & Projects

- **Professional results** with minimal effort
- **Consistent branding** across presentations
- **Rapid iteration** based on feedback
- **Easily updatable** as project evolves
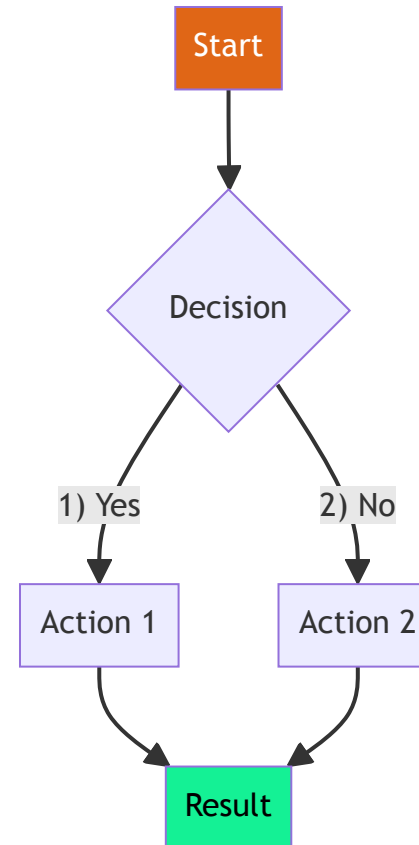- **Export to multiple formats** for different audiences

Time Savings: Reduce presentation creation time by 75% compared to traditional methods

Quality Improvement: Achieve professional design standards without design expertise

Technical Demo: Mermaid Diagrams

Mermaid Syntax Example

```
<div class="mermaid">
flowchart TD
    A[Start] --> B{Decision}
    B -->|Yes| C[Action 1]
    B -->|No| D[Action 2]
    C --> E[Result]
    D --> E
    style A fill:#9945FF,color:#FFF
    style E fill:#14F195,color:#000
</div>
```

- **Add directly in markdown**
- **Automatic rendering** in presentation
- **Perfect for architecture diagrams**

AI-Generated Diagrams: Use Cursor to create Mermaid diagrams from project descriptions

Practical Next Steps

Resources We'll Provide

- **GitHub repository** with all templates
- **AI prompt cheat sheet** for pitch decks
- **Marp theme template** with Solana styling
- **Sample Mermaid diagrams** for common use cases
- **Step-by-step tutorial** recording

Your Action Items

- **Install Cursor** (cursor.sh)
- **Set up VS Code** with Marp extension
- **Clone our GitHub repo** for templates
- **Gather your project materials**
- **Follow our workflow** for your next pitch

Build Projects, Not Presentations

Let AI and markdown do the heavy lifting

Contact: **info@gigentic.ai** | Repo: **github.com/csmarc/AI-Guild-Cursor-Marp-Slides**