

Project Charter & Scope

Nama Project: MCP Agents Laravel UI Generator
Sponsor: Internal Development Team
Project Manager: Fikri Armia Fahmi (2023071018)
Frontend Developer: Nadia (2024071004)
Tanggal Pembuatan: 18 September 2025
Status: Approved

1. Executive Summary

Project MCP Agents Laravel UI Generator adalah inisiatif untuk mengembangkan sistem AI multi-agent yang dapat mengotomatisasi pembuatan komponen UI Laravel dari natural language prompts. System ini dirancang untuk mempercepat development workflow, mengurangi repetitive tasks, dan meningkatkan produktivitas developer, khususnya mahasiswa dan developer pemula.

2. Project Purpose & Justification

2.1 Business Case

Pengembangan komponen UI di Laravel memerlukan pemahaman mendalam tentang Blade templating, routing, dan struktur project. Proses manual ini memakan waktu 2-4 jam per komponen untuk developer pemula, yang dapat dikurangi hingga 80% dengan otomatisasi AI.

2.2 Problem Statement

Terdapat beberapa permasalahan utama yang dihadapi oleh developer dalam pengembangan Laravel UI, yaitu:

- Developer pemula menghabiskan terlalu banyak waktu untuk boilerplate code
- Inkonsistensi dalam struktur dan naming conventions antar komponen
- Learning curve yang tinggi untuk Laravel best practices
- Repetitive tasks mengurangi waktu untuk fokus pada business logic

2.3 Opportunity

Dengan memanfaatkan AI multi-agent system dan Model Context Protocol (MCP), project ini memberikan peluang untuk mencapai beberapa keuntungan strategis sebagai berikut:

- Mengotomatisasi 70-80% proses pembuatan UI components
- Standardisasi code quality dan structure
- Mempercepat onboarding developer baru
- Meningkatkan produktivitas tim development

3. Project Objectives

3.1 Primary Objectives

Project ini memiliki tiga tujuan utama yang harus dicapai untuk memastikan kesuksesan implementasi, yaitu:

1. Otomatisasi Pembuatan UI

- Generate Blade components dari natural language prompts
- Otomatis membuat routes, controllers, dan validation rules
- Success metric: 90% generated code valid tanpa manual editing

2. Interactive Preview System

- Real-time preview dari generated components
- User dapat review dan modify sebelum integration
- Success metric: 95% user satisfaction dengan preview accuracy

3. Quality Assurance

- Automated validation untuk syntax, structure, dan best practices
- Linting dan acceptance criteria checking
- Success metric: Zero critical bugs dalam generated code

3.2 Secondary Objectives

Selain tujuan utama, project ini juga memiliki beberapa tujuan sekunder yang mendukung adopsi dan keberlanjutan sistem, meliputi:

- Dokumentasi lengkap dan user-friendly
- Example repository dengan 10+ use cases
- Community adoption minimal 50 users dalam 3 bulan pertama

4. Project Scope

4.1 In Scope

4.1.1 Core Features

System ini akan mengimplementasikan AI Agent System yang terdiri dari 10 agents dengan fungsi spesifik, yaitu:

1. Prompt Processing Agent - expand dan plan user prompts
2. Draft Generator - generate HTML preview
3. Layout Generator - create Blade layouts
4. Component Generator - create reusable Blade components
5. Route Generator - update routes/web.php
6. Controller Generator - create controller methods
7. Validation Generator - create form validation rules
8. Validator Agent - check syntax dan structure
9. Project Integrator - integrate files ke Laravel project
10. Documentation Generator - generate inline documentation

4.1.2 User Interface

Untuk memudahkan interaksi user dengan system, akan dikembangkan web-based interface dengan fitur-fitur berikut:

- Web-based preview interface (HTML/CSS/JS)
- Prompt input dengan syntax highlighting
- Interactive preview pane dengan live reload
- Agent status monitoring dashboard
- Accept/modify/reject controls
- Download generated files
- One-click integration ke Laravel project

4.1.3 Backend Services

Backend system akan menyediakan infrastruktur yang robust untuk menjalankan agents dan mengelola workflow, mencakup:

- RESTful API dengan FastAPI/Flask
- Asynchronous job queue (Celery/Redis)
- Job status tracking dan logging
- Error handling dan retry mechanism
- Rate limiting dan security controls

4.1.4 Integration

System akan terintegrasi dengan tools dan platform yang umum digunakan dalam Laravel development, meliputi:

- Laravel 8+ compatibility
- GitHub integration untuk PR creation
- Staging area untuk safe file operations
- Rollback mechanism

4.1.5 Documentation

Untuk mendukung adopsi dan penggunaan system, akan disediakan dokumentasi lengkap yang mencakup:

- User guide dan tutorials
- API documentation
- Architecture documentation
- Example repository dengan sample projects
- Video tutorials (optional)

4.2 Out of Scope

4.2.1 Tidak Termasuk dalam Project Ini

Untuk menjaga fokus dan memastikan delivery tepat waktu, beberapa fitur berikut tidak akan diimplementasikan dalam versi 1.0:

- Database migration generation (future enhancement)

- Model relationship generation (future enhancement)
- Testing code generation (future enhancement)
- Deployment automation (future enhancement)
- Multi-framework support (hanya Laravel untuk v1.0)
- Mobile app interface (web-only untuk v1.0)
- Real-time collaboration features
- Version control integration selain GitHub
- Custom AI model training (menggunakan existing Mistral API)

4.2.2 Assumptions

Project ini dibangun dengan beberapa asumsi dasar mengenai environment dan kemampuan users, yaitu:

- Users memiliki Laravel project yang sudah ter-setup
- Users memiliki basic knowledge tentang Laravel
- Internet connection tersedia untuk AI API calls
- Python 3.7+ dan Laravel 8+ sudah terinstall
- Users memiliki Mistral API access

4.2.3 Constraints

Terdapat beberapa batasan yang harus dipertimbangkan dalam pelaksanaan project ini, meliputi:

- Budget: Open-source project dengan minimal infrastructure cost
- Timeline: 14 minggu development (3.5 bulan)
- Resources: 2 developers (1 backend + PM, 1 frontend)
- Technology: Harus menggunakan Mistral API untuk AI inference
- Compatibility: Support Laravel 8+ only

5. Deliverables

5.1 Technical Deliverables

#	Deliverable	Description	Due Date
1	AI Agent Modules	10 Python modules untuk agent system	Week 11
2	Backend API	FastAPI/Flask REST API	Week 12
3	Frontend Preview UI	Web interface untuk preview dan control	Week 8
4	Integration System	File mover dan Laravel integrator	Week 10
5	Validation System	Syntax checker dan linter	Week 11
6	Queue System	Async job processing dengan Celery/Redis	Week 12

5.2 Documentation Deliverables

Dokumentasi yang akan diserahkan untuk mendukung penggunaan dan maintenance system adalah sebagai berikut:

#	Deliverable	Description	Due Date
1	User Guide	Step-by-step tutorial untuk end users	Week 13
2	API Documentation	REST API endpoints documentation	Week 13
3	Architecture Doc	System design dan architecture	Week 13
4	Example Repository	Sample Laravel projects dengan generated UI	Week 13
5	README	Installation dan quick start guide	Week 13

5.3 Project Management Deliverables

Untuk keperluan project management dan knowledge transfer, akan disediakan dokumen-dokumen berikut:

- Weekly sprint reports
- Monthly stakeholder presentations
- Final project handover document
- Lessons learned document

6. Stakeholders

6.1 Internal Stakeholders

Stakeholder internal yang terlibat langsung dalam pelaksanaan project ini adalah sebagai berikut:

Role	Name	Responsibility	Involvement
Project Sponsor	Internal Dev Team	Funding dan strategic direction	High
Project Manager	Fikri Armia Fahmi	Overall project management, backend dev	High
Frontend Developer	Nadia	UI/UX development	High
QA Team	TBD	Testing dan quality assurance	Medium

6.2 External Stakeholders

Stakeholder eksternal yang memiliki kepentingan terhadap kesuksesan project ini meliputi:

Stakeholder	Interest	Influence	Engagement Strategy
Laravel Developers	End users dari tool	High	Beta testing, feedback sessions
Software Houses	Potential adopters	Medium	Demo sessions, case studies
Open Source Community	Contributors	Medium	GitHub issues, documentation
Students	Learning dan usage	Low	Tutorials, educational content

7. Success Criteria

7.1 Technical Success Metrics

Kesuksesan teknis project akan diukur berdasarkan beberapa metrik kinerja berikut:

- ☒ 90% generated Blade components syntactically valid
- ☒ Preview UI load time < 2 seconds
- ☒ API response time < 500ms untuk simple requests
- ☒ Zero data loss during integration process
- ☒ Support untuk 10+ common UI patterns (forms, tables, cards, etc.)

7.2 Business Success Metrics

Dari perspektif bisnis, kesuksesan project akan dievaluasi melalui indikator-indikator berikut:

- ☒ 50+ active users dalam 3 bulan pertama
- ☒ 80% user satisfaction score
- ☒ 3+ case studies dari real projects
- ☒ 100+ GitHub stars dalam 6 bulan
- ☒ Average time saving 60%+ per component

7.3 Quality Metrics

Kualitas deliverables akan dipastikan melalui pencapaian metrik-metrik berikut:

- ☒ Code coverage > 70% untuk critical modules
- ☒ Zero critical security vulnerabilities
- ☒ Documentation completeness > 90%
- ☒ Bug resolution time < 48 hours untuk critical issues

8. Project Timeline & Milestones

8.1 High-Level Timeline

Total Duration: 14 minggu (18 September 2025 - 25 Desember 2025)

8.2 Major Milestones

Milestone	Description	Target Date	Deliverables
M1: Project Kickoff	Setup infrastructure dan repo	Week 2	Repo, CI/CD, env setup
M2: Core Agents Ready	Prompt processor dan draft generator	Week 6	2 working agents
M3: Preview UI Alpha	Basic preview interface	Week 8	Functional preview UI
M4: Integration Complete	Full agent pipeline working	Week 10	End-to-end workflow
M5: Validation System	Quality checks implemented	Week 11	Validator agent
M6: Beta Release	Feature complete, testing phase	Week 12	Beta version

Milestone	Description	Target Date	Deliverables
M7: Documentation	All docs completed	Week 13	Complete documentation
M8: Production Release	v1.0 launch	Week 14	Production-ready system

8.3 Sprint Breakdown

Development akan dilakukan secara iteratif dengan pembagian sprint sebagai berikut:

- **Sprint 1-2 (Week 1-2):** Infrastructure setup
- **Sprint 3-4 (Week 3-4):** Prompt processing agents
- **Sprint 5-6 (Week 5-6):** Draft generator dan preview scaffold
- **Sprint 7-8 (Week 7-8):** Layout/component generator + UI enhancements
- **Sprint 9-10 (Week 9-10):** Route agent dan integrator
- **Sprint 11 (Week 11):** Validator agent
- **Sprint 12 (Week 12):** Backend polishing
- **Sprint 13 (Week 13):** Documentation
- **Sprint 14 (Week 14):** Final testing dan release

9. Budget & Resources

9.1 Human Resources

Role	Allocation	Duration	Cost (Estimated)
Project Manager + Backend Dev	100%	14 weeks	Internal resource
Frontend Developer	100%	14 weeks	Internal resource
QA Support	25%	4 weeks	Internal resource

9.2 Infrastructure Costs

Item	Monthly Cost	Duration	Total
Mistral API Credits	\$50	4 months	\$200
Cloud Server (optional)	\$20	4 months	\$80
GitHub (free tier)	\$0	-	\$0
Domain (optional)	\$12/year	1 year	\$12
Total Estimated Cost			\$292

9.3 Tools & Software

Tools dan software yang akan digunakan dalam development project ini meliputi:

- Python 3.7+ (free)
- Laravel 8+ (free)
- FastAPI/Flask (free)
- Redis/Celery (free)
- GitHub (free tier)
- VS Code / IDE (free)
- Mistral API (paid)

10. Risk Management

10.1 High Priority Risks

Risiko-risiko dengan prioritas tinggi yang telah diidentifikasi dan memerlukan perhatian khusus adalah:

Risk	Probability	Impact	Mitigation Strategy
Mistral API downtime	Medium	High	Implement retry logic, cache templates, graceful degradation
Generated code breaks Laravel project	Medium	High	Staging area, branch creation, rollback mechanism
Scope creep	High	Medium	Strict scope management, change control process
Resource unavailability	Low	High	Cross-training, documentation, backup resources

10.2 Medium Priority Risks

Risiko-risiko dengan prioritas menengah yang perlu dimonitor secara berkala meliputi:

Risk	Probability	Impact	Mitigation Strategy
Security vulnerabilities	Medium	Medium	Code review, security scanning, input sanitization
Performance issues	Medium	Medium	Load testing, optimization, caching
User adoption low	Medium	Medium	Marketing, tutorials, community engagement
Integration complexity	High	Low	Incremental integration, extensive testing

11. Communication Plan

11.1 Internal Communication

Komunikasi internal tim akan dilakukan melalui beberapa forum regular berikut:

- **Daily Standups:** 15 menit setiap pagi (async via chat)
- **Weekly Sprint Review:** Setiap Jumat, 1 jam
- **Sprint Planning:** Setiap Senin, 1 jam
- **Monthly Stakeholder Update:** Presentasi progress dan demo

11.2 External Communication

Untuk komunikasi dengan stakeholder eksternal dan community, akan digunakan channel-channel berikut:

- **GitHub Issues:** Public issue tracking
- **Documentation Site:** User guides dan tutorials
- **Blog Posts:** Monthly updates tentang progress
- **Social Media:** Announcements di Twitter/LinkedIn (optional)

11.3 Reporting

Pelaporan progress dan status project akan dilakukan secara terstruktur melalui:

- Weekly sprint reports (internal)
- Monthly progress reports (stakeholders)
- Incident reports (as needed)
- Final project report (end of project)

12. Quality Management

12.1 Quality Standards

Untuk memastikan kualitas deliverables, project ini akan mengikuti standar-standar berikut:

- Code harus follow PEP 8 (Python) dan PSR-12 (PHP)
- All public APIs harus documented
- Critical functions harus memiliki unit tests
- Security best practices harus diikuti
- Accessibility standards (WCAG 2.1 Level A minimum)

12.2 Quality Assurance Process

Proses quality assurance akan dilakukan melalui beberapa tahapan verifikasi sebagai berikut:

1. **Code Review:** Semua code changes harus di-review
2. **Automated Testing:** CI/CD pipeline dengan automated tests
3. **Manual Testing:** QA testing untuk critical features
4. **Security Scanning:** Automated security vulnerability scanning
5. **Performance Testing:** Load testing untuk API endpoints

12.3 Acceptance Criteria

Setiap deliverable harus memenuhi kriteria penerimaan berikut sebelum dianggap complete:

- All features harus memenuhi acceptance criteria di requirements
- Zero critical bugs dalam production
- Documentation completeness verified
- User acceptance testing passed
- Performance benchmarks met

13. Change Management

13.1 Change Control Process

Setiap perubahan terhadap scope, timeline, atau resources harus melalui proses formal berikut:

- 1. **Change Request:** Submit via GitHub issue atau formal request
- 2. **Impact Analysis:** Assess impact pada scope, timeline, budget
- 3. **Approval:** PM dan sponsor review dan approve/reject
- 4. **Implementation:** Jika approved, masuk ke sprint backlog
- 5. **Communication:** Notify stakeholders tentang changes

13.2 Scope Change Approval Authority

Authority untuk menyetujui perubahan scope ditentukan berdasarkan magnitude perubahan sebagai berikut:

- **Minor changes** (< 1 day effort): PM approval
- **Medium changes** (1-3 days effort): PM + Sponsor approval
- **Major changes** (> 3 days effort): Full stakeholder review

14. Project Closure Criteria

14.1 Completion Criteria

Project akan dianggap selesai dan siap untuk closure apabila semua kriteria berikut telah terpenuhi:

- ☒ All deliverables completed dan accepted
- ☒ All acceptance criteria met
- ☒ Documentation complete dan published
- ☒ Production deployment successful
- ☒ User training completed (if applicable)
- ☒ Handover document signed off
- ☒ Lessons learned documented

14.2 Post-Project Activities

Setelah project closure, beberapa aktivitas berikut akan dilakukan untuk memastikan keberlanjutan system:

- Maintenance plan established
- Support process defined
- Enhancement backlog created
- Community management plan
- Success metrics monitoring setup

15. Approval & Sign-off

15.1 Document Approval

Role	Name	Signature	Date
Project Sponsor	Internal Dev Team	_____	_____
Project Manager	Fikri Armia Fahmi	_____	18/09/2025

Role	Name	Signature	Date
Frontend Developer	Nadia	_____	18/09/2025

15.2 Charter Authorization

This project charter authorizes the Project Manager to proceed with the project and allocate resources as outlined in this document.

Approved by:
Sponsor: _____
Date: _____

16. Appendices

Appendix A: Glossary

Berikut adalah definisi istilah-istilah teknis yang digunakan dalam dokumen ini:

- **MCP:** Model Context Protocol
- **Blade:** Laravel's templating engine
- **Agent:** Autonomous AI component dengan specific task
- **FastAPI:** Modern Python web framework
- **Celery:** Distributed task queue
- **Redis:** In-memory data store

Appendix B: References

Referensi dan dokumentasi yang relevan untuk project ini meliputi:

- Laravel Documentation: <https://laravel.com/docs>
- Mistral API Documentation: <https://docs.mistral.ai>
- FastAPI Documentation: <https://fastapi.tiangolo.com>
- Project Repository: [TBD]

Appendix C: Contact Information

Informasi kontak untuk stakeholder utama project adalah sebagai berikut:

- **Project Manager:** Fikri Armia Fahmi - [email]
- **Frontend Developer:** Nadia - [email]
- **Sponsor Contact:** [email]

Document Version: 1.0
Last Updated: 18 September 2025
Next Review Date: 2 Oktober 2025