NexusForge AI

v1.0.1

Complete End User Guide

"You Direct. NexusForge Builds. Zero Code Required."

Document Version	1.0.1
Product Version	1.0.1
Classification	End User Documentation

Table of Contents

1. Getting Started

What is NexusForge AI?

NexusForge AI is an autonomous development environment that transforms your ideas into working software using natural language. Simply describe what you want to build, and NexusForge creates complete, production-ready applications.

Key Benefits

- **100% Local Operation**: Your code never leaves your machine
- **Autonomous Development**: Minimal manual coding required
- **Privacy-First**: Complete control over your data
- **Multi-Domain**: Supports any technology stack
- **Enterprise-Grade**: Professional quality with safety guarantees

Who Should Use NexusForge?

- **Developers**: Accelerate development with AI assistance
- **Product Managers**: Rapidly prototype and validate ideas
- **Entrepreneurs**: Build MVPs without extensive technical knowledge
- **Students**: Learn by example with AI-generated code
- **Teams**: Standardize development practices and workflows

2. Installation & Setup

Prerequisites

- **Operating System**: Linux, macOS, or Windows
- **Python**: 3.8 or higher
- **Memory**: 8GB RAM minimum (16GB recommended)
- **Storage**: 10GB free space
- **GPU**: Optional but recommended for faster AI processing

Installation Steps

Step 1: Install NexusForge

```
git clone https://github.com/nexusforge-ai/nexusforge.git
```

```
cd nexusforge && pip install -e .
```

Step 2: Install Ollama (Local AI)

```
curl -fsSL https://ollama.ai/install.sh | sh
```

Step 3: Initialize NexusForge

```
nexusforge init
```

Step 4: Verify Installation

```
nexusforge status
```

You should see: NexusForge AI Ready, Ollama Connected, Models Available, Monitoring Active

3. First Steps

Your First Project

1. Start a Chat Session

```
nexusforge chat
```

2. Describe Your Project

Example: "I want to build a todo app with React and Node.js that includes user authentication and real-time updates."

3. Let NexusForge Analyze

NexusForge will analyze your requirements, suggest architecture, recommend technologies, and create a project plan.

4. Approve and Build

Once you confirm, NexusForge will autonomously create project structure, generate all files, set up database schemas, implement authentication, add real-time features, create tests, and generate documentation.

Understanding the Output

- frontend/ - React application
- backend/ - Node.js API
- database/ - Database setup
- tests/ - Comprehensive test suite
- docs/ - Complete documentation
- docker-compose.yml - Easy deployment
- README.md - Setup instructions

4. Core Concepts

1. Autonomous Building

NexusForge doesn't just generate code—it creates complete, working applications:

- **Architecture Planning**: Designs optimal system architecture
- **Technology Selection**: Chooses best-fit technologies
- **Code Generation**: Writes production-quality code
- **Testing**: Creates comprehensive test suites
- **Documentation**: Generates complete documentation

2. Natural Language Programming

Describe what you want in plain English:

- "Build a web app for managing customer orders"
- "Create a mobile app that tracks fitness goals"
- "Design a microservice for processing payments"
- "Build a dashboard for monitoring server metrics"

3. Context Awareness

- **Project History**: Remembers all previous changes
- **Code Relationships**: Understands how files connect
- **Technology Stack**: Knows your chosen frameworks
- **Best Practices**: Follows established patterns

4. Safety-First Approach

- **Operation Validation**: Confirms risky operations
- **Automatic Backups**: Saves versions before changes
- **Rollback Capability**: Undo any unwanted changes
- **Security Scanning**: Detects potential vulnerabilities

5. Basic Usage

Core Commands

Command	Description
`nexusforge chat`	Interactive conversation mode
`nexusforge init [project-name]`	Quick project setup
`nexusforge build [options]`	Autonomous building
`nexusforge status`	Project status
`nexusforge prd load <file>`	Load requirements document
`nexusforge direct "<instruction>"`	Direct high-level instruction
`nexusforge task "<task>"`	Execute specific task

Build Options

- **--interactive**: Step-by-step guided building
- **--autonomous**: Fully autonomous execution
- **--aggressive**: Fast iteration mode
- **--lightning**: Maximum speed building

Slash Commands in Chat Mode

- /prd load - Load and analyze PRD
- /plugins list - Show available plugins
- /plugins install - Install community plugin
- /monitor open - Launch monitoring dashboard
- /session save - Save current session
- /session load - Load saved session
- /build start - Begin autonomous build
- /help - Show all commands

6. Advanced Features

Autonomous Build Modes

Interactive Mode

```
nexusforge build --interactive
```

Best for learning. Provides step-by-step explanations, confirmation before each phase, educational insights, and customization options.

Autonomous Mode

```
nexusforge build --autonomous
```

For experienced users. Minimal intervention, optimal architecture decisions, best practice implementation, and comprehensive error handling.

Lightning Mode

```
nexusforge build --lightning
```

Maximum speed for rapid prototyping. Parallel file generation, minimal validation, quick MVP creation, and iterative improvement.

Project Templates

- **Web**: react-app, nextjs-fullstack, vue-spa
- **APIs**: fastapi-rest, express-graphql, django-rest
- **Mobile**: react-native, flutter-app
- **Desktop**: electron-app, tauri-app

```
nexusforge init --template react-app
```

7. Project Management

Project Initialization

When you run nexusforge init, NexusForge performs deep analysis:

- **Directory Analysis**: Understands existing code structure
- **Dependency Detection**: Identifies all dependencies and frameworks
- **Pattern Recognition**: Learns your coding patterns
- **Configuration Setup**: Creates optimal configuration
- **Documentation Generation**: Creates comprehensive docs

Requirements Management

```
nexusforge prd load requirements.md
```

NexusForge analyzes requirements for:

- Feature Completeness: Identifies missing requirements
- Technical Feasibility: Assesses implementation complexity
- Architecture Implications: Suggests optimal architecture
- Timeline Estimation: Provides realistic timelines
- Risk Assessment: Identifies potential challenges

Project Status Monitoring

```
nexusforge status
```

- Project Health: Overall project status
- Build Status: Last build results
- Test Coverage: Code coverage metrics
- Technical Debt: Code quality issues
- Security: Security scan results

8. AI Interaction Patterns

Effective Communication

Being Specific

> Poor: "Make a website"

> Good: "Create a portfolio website with a hero section, project gallery, contact form, and blog using React and Tailwind CSS"

Iterative Development

Start simple and add features iteratively:

1. "Create a basic todo app with add and delete functionality"
2. "Add categories and priority levels to the todos"
3. "Add user accounts and todo sharing"
4. "Add email reminders and notifications"

Advanced Interaction Techniques

Architecture Discussions

"What's the best architecture for a real-time chat application that needs to scale to 10,000+ concurrent users?"

Code Review

"Review the authentication middleware for security issues and performance problems"

Debugging Assistance

"The payment processing is failing with a 400 error, but I can't figure out why"

9. Plugin Management

Plugin Tiers

Tier	Plugins	Price
Free	1 community plugin	$0
Solo Pro	5 community plugins	$19/month
Team	20 community plugins	$49/month
Enterprise	Unlimited	Custom

Plugin Commands

- nexusforge plugins list - View all available plugins
- nexusforge plugins info - Show plugin details
- nexusforge plugins install - Install plugin
- nexusforge plugins remove - Remove plugin
- nexusforge plugins update - Update all plugins

Core Plugins (Always Available)

- **AI/ML**: Model management, prompt engineering, RAG
- **Database**: PostgreSQL, MongoDB, Redis, SQLite
- **IoT**: Arduino, ESP32, Raspberry Pi
- **Security**: OAuth, JWT, encryption, scanning
- **Frameworks**: React, Vue, Django, FastAPI, Flutter

10. Security & Privacy

Privacy-First Design

What Data Stays Local:

- Source Code: Never transmitted outside your machine
- Project Files: Stored only on your local system
- Conversations: Kept in local sessions
- Credentials: Stored in secure local keyring
- Personal Information: Never collected or transmitted

Security Features

Path Protection:

NexusForge prevents modifying system directories (/etc, /usr, /bin), accessing sensitive files, and operations outside project boundaries.

Command Validation:

Dangerous commands are blocked: rm -rf /, sudo chmod 777, git reset --hard HEAD~100

Privacy Controls

- nexusforge telemetry status - Check telemetry status
- nexusforge telemetry disable - Disable all telemetry
- nexusforge data show - Show all stored data
- nexusforge data purge --confirm - Complete data removal

11. Performance Optimization

Monitoring Dashboard

```
nexusforge monitor
```

```
Dashboard available at: http://localhost:5555
```

Dashboard shows:

- System Metrics: CPU, RAM, GPU usage
- AI Performance: Response times, token usage
- Cache Efficiency: Hit rates, compression ratios
- Project Statistics: Build times, file counts

Performance Commands

- nexusforge perf status - Performance status
- nexusforge perf benchmark - Benchmark system
- nexusforge perf optimize - Optimize for current project
- nexusforge compress benchmark - Test compression
- nexusforge cache optimize - Optimize cache settings

OpenZL Compression Benefits

- 90% space reduction for project files
- Sub-millisecond file access after caching
- Intelligent learning from your project patterns

12. Troubleshooting

Common Issues

Python Version Issues

```
python --version # Should be 3.8 or higher
```

Ollama Connection Issues

```
ollama list && ollama serve
```

Permission Errors (Linux/macOS)

```
sudo chown -R $USER:$USER ~/.nexusforge/
```

AI Responses Slow

- nexusforge perf optimize
- nexusforge config set model lightweight-model
- Enable GPU acceleration: nexusforge gpu configure

Cache Corruption

```
nexusforge cache clear && nexusforge cache rebuild
```

Getting Help

- nexusforge help - General help
- nexusforge help - Command-specific help
- nexusforge diagnose - Full system diagnostic
- nexusforge logs show - View recent logs

13. Best Practices

Project Organization

Recommended folder structure:

- .nexusforge/ - NexusForge configuration
- src/ - Source code (components, pages, utils, config)
- tests/ - Test suites (unit, integration, e2e)
- docs/ - Documentation
- scripts/ - Build and utility scripts

Communication Best Practices

> Good: "Create a user registration form with email validation, password strength checking, terms of service agreement, and success/error messaging"

> Poor: "Make a signup form"

Security Best Practices

- Always use secure authentication with password hashing and JWT
- Add comprehensive input validation for all API endpoints
- Use environment variables for all secrets and API keys
- Request security scanning after major changes

14. Keyboard Shortcuts

Global Shortcuts

Shortcut	Action
Ctrl+C / Cmd+C	Cancel current operation
Ctrl+D / Cmd+D	Exit chat mode
Ctrl+L / Cmd+L	Clear screen
Ctrl+R / Cmd+R	Refresh/reload
Tab	Auto-complete command/path

Chat Mode Shortcuts

- Up/Down arrows: Navigate command history
- Ctrl+A / Cmd+A: Move to beginning of line
- Ctrl+E / Cmd+E: Move to end of line
- Ctrl+W / Cmd+W: Delete word backwards
- Ctrl+U / Cmd+U: Clear entire line

Navigation Shortcuts

- Ctrl+N / Cmd+N: New project/session
- Ctrl+O / Cmd+O: Open project
- Ctrl+S / Cmd+S: Save current session
- Ctrl+P / Cmd+P: Open quick command palette

15. Configuration Reference

Configuration File Location

Linux/macOS: ~/.nexusforge/config.yaml | Windows: %USERPROFILE%\.nexusforge\config.yaml

Basic Configuration

- **AI Settings**: provider, model, temperature, max_tokens
- **Build Settings**: mode, auto_test, auto_docs, quality_gates
- **Performance**: cache_enabled, compression_enabled, gpu_acceleration
- **Security**: safe_mode, confirm_risky_operations, backup_before_changes
- **Privacy**: telemetry_enabled, crash_reporting, usage_analytics

Environment Variables

- NEXUSFORGE_HOME - Configuration directory
- NEXUSFORGE_MODEL - Default AI model
- NEXUSFORGE_CACHE_SIZE - Cache size limit
- NEXUSFORGE_GPU_ENABLED - Enable GPU acceleration
- NEXUSFORGE_LOG_LEVEL - Logging verbosity

16. FAQ

General Questions

Q: Is my code sent to external servers?

> A: No. NexusForge operates 100% locally. Your code never leaves your machine.

Q: What programming languages does NexusForge support?

A: All major languages: JavaScript, TypeScript, Python, PHP, Ruby, Swift, Kotlin, Dart, C#, Java, Rust, Go, C++, R, SQL, and more.

Q: Can I use NexusForge for commercial projects?

A: Yes. The generated code is yours to use commercially without restrictions.

Technical Questions

Q: How much RAM does NexusForge need?

A: 8GB minimum, 16GB recommended, 32GB for large projects.

Q: Does NexusForge work offline?

> A: Yes. Once models are downloaded, NexusForge works completely offline.

Q: Does NexusForge generate tests?

A: Yes. Comprehensive test suites including unit, integration, e2e, and security tests.

Getting Support

- Official Docs: https://docs.nexusforge.ai
- Discord: https://discord.gg/nexusforge
- GitHub: https://github.com/nexusforge-ai
- Email: support@nexusforge.ai

NexusForge AI transforms the way you build software by providing an intelligent, autonomous development partner that understands your requirements and delivers production-ready solutions.