Loomix User Manual

Version: 1.0.0

Date: November 2025

Product: Loomix AI Code Review Platform

1. Introduction

Welcome to **Loomix**, the next-generation automated code review infrastructure designed to weave quality into every line of your codebase. Loomix combines static analysis with advanced AI models to detect bugs, security vulnerabilities, and performance bottlenecks before they reach production.

Key Benefits

- **Speed:** Analyze thousands of lines of code in seconds.
- **Precision:** AI-driven context awareness reduces false positives.
- **Integration:** Seamlessly connects with GitHub, GitLab, and Bitbucket.

2. Installation

Loomix can be installed as a CLI tool for local development or integrated directly into your CI/CD pipeline.

System Requirements

- **Node.js:** v18.0.0 or higher
- **OS:** Linux, macOS, or Windows (WSL2 recommended)
- **RAM:** Minimum 4GB (8GB recommended for large repositories)

CLI Installation

To install the Loomix CLI globally on your machine:

```
npm install -g @loomix/cli
```

Verify the installation:

```
loomix --version
```

3. Getting Started

Authentication

Before analyzing code, you must authenticate with the Loomix cloud service to access premium AI models.

```
loomix auth login
```

Follow the on-screen prompts to authorize via your browser.

Connecting a Repository

Navigate to your project directory and initialize Loomix:

```
cd my-project
loomix init
```

This will create a `.loomixrc` configuration file in your project root.

Running Your First Scan

To scan your current branch for issues:

```
loomix scan
```

4. Core Features

4.1 The Loom Engine

The **Loom Engine** is the heart of Loomix. It parses your code into an Abstract Syntax Tree (AST) and applies over 500 predefined rule sets for best practices in JavaScript, TypeScript, Python, and Go.

4.2 AI Analysis

Unlike traditional linters, Loomix uses Large Language Models (LLMs) to understand the *intent* of your code.

- **Logic Error Detection:** Finds infinite loops, race conditions, and off-by-one errors that static analysis misses.
- **Refactoring Suggestions:** Proposes cleaner, more readable alternatives to complex functions.

4.3 Pull Request Integration

Loomix automatically comments on Pull Requests (PRs).

- **Critical Issues:** Blocks merging until resolved.
- **Nitpicks:** Posted as "suggestions" that can be applied with a single click.

5. Configuration

Loomix is highly configurable via the `.loomixrc` (JSON or YAML) file.

Example `.loomixrc`:

```
{
  "extends": "loomix:recommended",
  "rules": {
    "no-console": "warn",
    "complexity": ["error", { "max": 10 }],
    "security/detect-secrets": "error"
  },
  "ai": {
    "model": "loom-gpt-4",
    "creativity": "low"
  }
}
```

6. Troubleshooting

Common Issues

Error: "Authentication Failed"

- Check your internet connection.
- Run `loomix auth logout` followed by `loomix auth login`.

Error: "Scan Timeout"

- For very large repositories (>1M lines), increase the timeout setting in `.loomixrc`:

```
"settings": {
  "timeout": 600000
}
```

7. Support

For enterprise support, billing inquiries, or feature requests, please contact our dedicated support team.

- **Email:** support@loomix.dev
- **Documentation:** https://docs.loomix.dev

- **Status Page:** https://status.loomix.dev