

The **Ezy Robots system** operates as an automated diagnostic and corrective framework for analyzing and refining a website's `robots.txt` configuration. It first retrieves the existing file from the site's root directory using an adaptive fetch process that handles redirects, user-agent normalization, and timeout control. Once retrieved, the file is parsed and evaluated through a structured scoring system that detects AI crawler blocks (e.g., GPTBot, ClaudeBot, Google-Extended), conflicting or redundant directives, invalid syntax, and missing or duplicate sitemap entries. Each issue is classified by severity and category (AI accessibility, SEO compliance, or syntax integrity) to quantify its impact on overall crawl visibility. When optimization is enabled, Ezy's generative layer uses a fine-tuned GPT-4.1 model to reconstruct the file according to best practices—removing harmful disallow rules, restoring sitemap visibility, ensuring consistent path formatting, and explicitly allowing legitimate AI crawlers while maintaining protections for sensitive directories such as `/cart`, `/checkout`, and `/admin`. The system then performs structural validation to eliminate duplicate or contradictory entries, enforce directive ordering, and verify compliance with the Robots Exclusion Protocol. The result is a validated and AI-accessible `robots.txt` file optimized for both traditional search engine indexing and large language model (LLM) discovery, without compromising site security or privacy boundaries.

During the optimization phase, **Ezy Robots** employs a multi-stage generative pipeline that transforms analytical findings into a syntactically valid and crawler-optimized `robots.txt` file. The process begins with the structured output of the diagnostic layer—each detected issue (e.g., AI crawler block, sitemap omission, or rule duplication) is converted into a normalized JSON schema containing directive type, location, and remediation logic. This schema is then passed to a controlled GPT-4.1 generation environment that has been fine-tuned with a domain-specific prompt template encompassing RFC-9309 compliance, AI crawler taxonomies, and SEO best practices. The model synthesizes a new ruleset that maximizes accessibility for compliant agents while preserving restrictions for sensitive or transactional endpoints. It dynamically builds hierarchical user-agent sections, first generating a universal `User-agent: *` block and subsequently inserting agent-specific overrides for GPTBot, ClaudeBot, PerplexityBot, Google-Extended, and Bingbot. Each rule is validated against a lexical pattern engine that enforces normalized path syntax, consistent casing, and correct newline spacing. After generation, a deterministic validator re-parses the proposed file, ensuring that no contradictory `Allow/Disallow` pairs exist and that the final `Sitemap:` directive is uniquely positioned at the end of the file. A diff-tracking module then compares the new version with the original, tagging each modification by type (added, removed, or reorganized) and annotating it with rationale (e.g., *“AI access restored”*, *“duplicate disallow collapsed”*). The process concludes with a checksum verification to guarantee integrity and a confidence score indicating expected improvement in crawl accessibility and AI discoverability.