

Bounteous AI Hackathon

Ctrl-AI-Hack

ReviewGenie

June 2-6

Agenda

- Problem
 - Manual code reviews are time-consuming, inconsistent, and prone to missing security/code smells.
 - Post-release defects and delays due to missed issues in PRs.
- Impact
 - **Reduced Review Time:** Faster approvals and reduced wait times for developers.
 - **Consistency:** Ensures objective, standard-compliant reviews.
 - **Higher Quality:** Identifies security risks and code smells proactively.
 - **Fewer Post-Release Defects:** Reduces the risk of bugs and vulnerabilities reaching production.
- Solution
 - **AI-driven Tool:** Flags security issues, code smells, and enforces org-specific guidelines.
 - **Customizable:** Aligns with company-specific coding standards.
 - **Faster Approvals:** Automated suggestions for quick fixes and better code quality.

Problem / Impact

Problem

- Manual code reviews are slow, inconsistent, and prone to missing security vulnerabilities and code issues. This leads to delays in approvals, inconsistent code quality, and potential defects post-release.

Target Audience

- **Software Developers:** Looking to streamline time-consuming manual code reviews.
- **Engineering Teams:** Aiming to improve code quality, consistency, and reduce security risks.
- **DevOps Teams:** Interested in integrating automation into CI/CD pipelines for efficiency.
- **Tech Leads & Managers:** Focused on boosting productivity, reducing technical debt, and speeding up releases.
- **Security Experts:** Ensuring secure code and preventing vulnerabilities in production.

Impact

- **Cost Savings:** Reduces manual review time, improving developer productivity and lowering operational costs.
- **Faster Time to Market:** Speeds up PR approvals, enabling quicker feature deployment and better customer satisfaction.
- **Improved Quality:** Fewer defects and vulnerabilities, ensuring a more stable product.
- **Innovation:** AI-driven reviews offer a unique, scalable solution that differentiates the organization.

\$100K

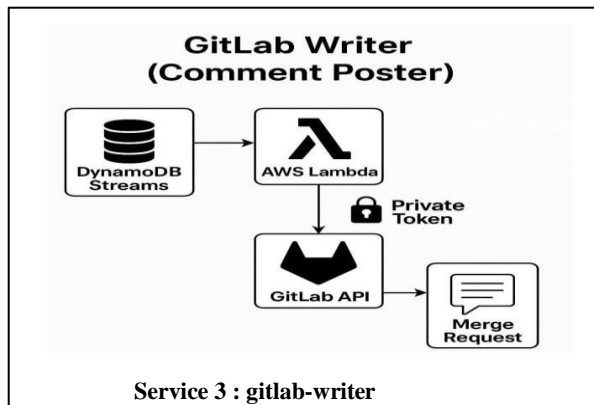
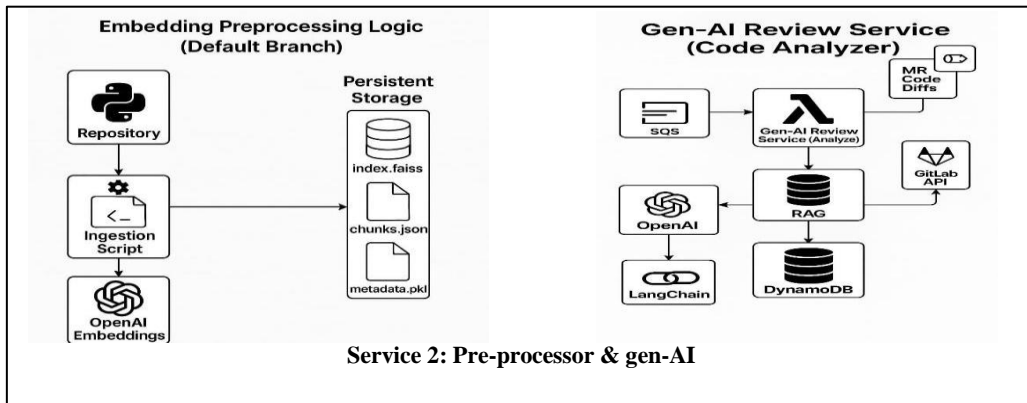
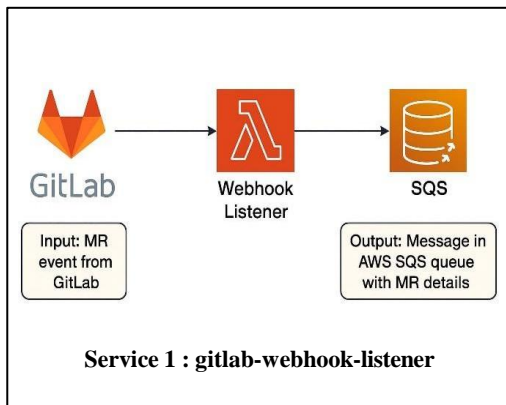
Saving / Revenue
etc

Solution

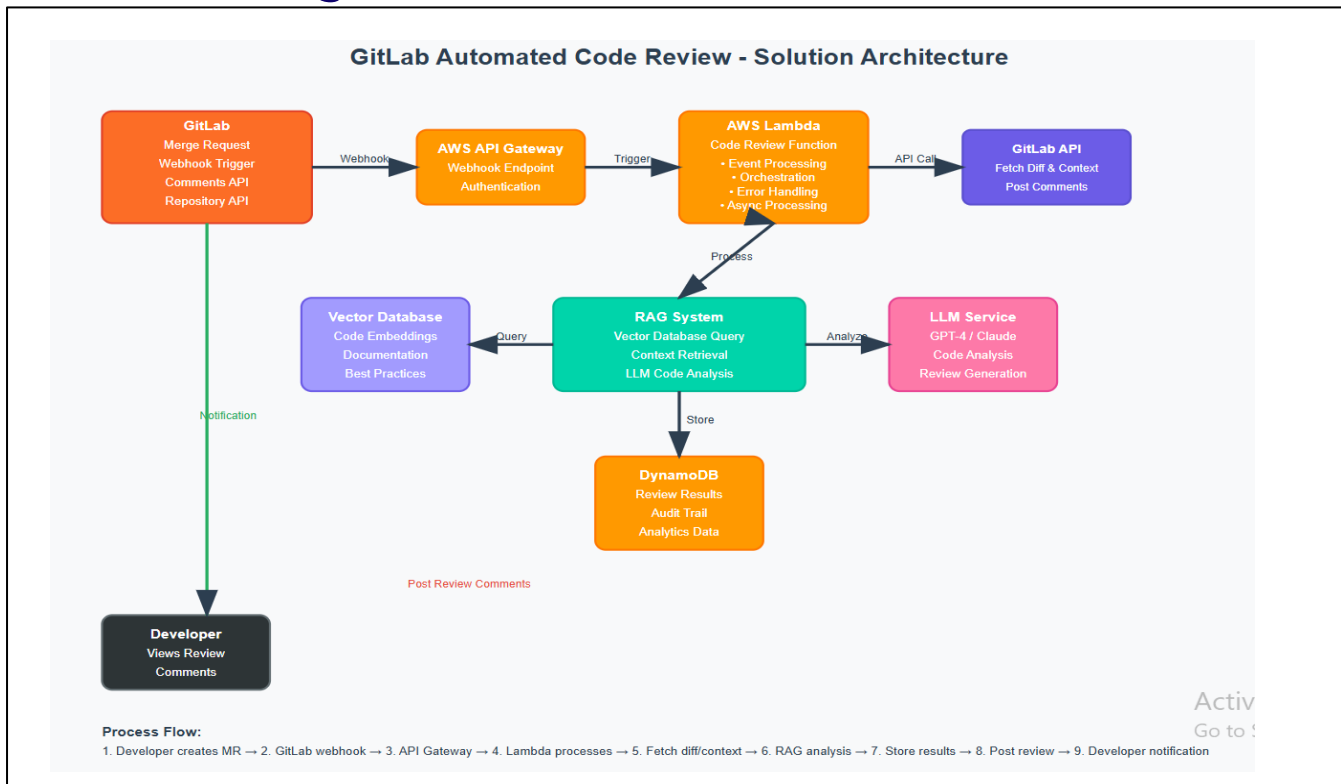
An LLM-based tool that integrates into your GitLab and triggers on Merge Request to:

- **Automated Code Reviews:** Automatically scans every pull request to flag security vulnerabilities, code smells, and deviations from organizational coding guidelines.
- **Customizable Guidelines:** Tailor the tool's review process to align with your organization's specific coding practices and standards, ensuring it enforces best practices that matter most to your team.
- **Security Vulnerability Detection:** Flag potential security risks in the code (such as SQL injections, XSS vulnerabilities, etc.) based on predefined rules and patterns.
- **Code Smell Detection:** Identify suboptimal code structures, such as duplicate code, unused variables, and overly complex functions, that could lead to future maintenance challenges.
- **Suggested Improvements:** Provide suggestions for code improvements, such as refactoring opportunities, performance enhancements, and more concise code alternatives.
- **Faster Approvals:** By automatically flagging and suggesting fixes for common issues, PR reviews can be completed much faster, improving developer productivity and reducing friction in the review process.
- **Easy Integration:** Seamlessly integrates with common code hosting platforms (GitHub, GitLab, Bitbucket, etc.), requiring minimal setup to get started.

Technical Architecture



Solution Design



Screenshot of working solution

• Merge Request Diff

Open Add realm to HTTP Basic Authentication add-mandatory-realm-on-bas... into master

Overview 1 Commits 1 Pipelines 0 Changes 3 Add a to-do item

Files 3

Search (e.g. *.vue) (Ctrl+P)

fastapi/security

http.py +3 -5

tests

test_tutorial/test_security

test_tutorial006.py +3 -3

test_secu...optional.py +2 -2

fastapi/security/http.py +3 -5 Viewed

```

184 - self.realm = realm
184 + self.realm = realm or ""
185 self.auto_error = auto_error
186
187 async def __call__( # type: ignore
@@ -189,10 +189,8 @@ class HTTPBasic(HTTPBase):
189 ) -> Optional[HTTPBasicCredentials]:
190 authorization = request.headers.get("Authorization")
191 scheme, param =
get_authorization_scheme_param(authorization)
192 if self.realm:
193 unauthorized_headers = {"WWW-Authenticate": f'Basic
realm={self.realm}'}
194 else:
195 unauthorized_headers = {"WWW-Authenticate": "Basic"}
192 + # The "realm" is required, as per
https://datatracker.ietf.org/doc/html/rfc7617#section-2.
193 + unauthorized_headers = {"WWW-Authenticate": f'Basic
realm={self.realm}'}

```

• ReviewGenie Analysis on Merge Request

Open Add realm to HTTP Basic Authentication add-mandatory-realm-on-bas... into master

Overview 1 Commits 1 Pipelines 0 Changes 3

Navya @Navya730 16 hours ago

Owner

High-Level Diff Analysis

The diff shows a change in the `HTTPBasic` class within the `fastapi/security/http.py` file. The primary modification is the handling of the `realm` attribute. Previously, the `realm` was directly assigned, and the `WWW-Authenticate` header was conditionally set based on its presence. The updated code assigns an empty string to `realm` if it is not provided and always includes the `realm` in the `WWW-Authenticate` header.

Security Issue Detection

- 1. Realm Requirement:** The change enforces the inclusion of a `realm` in the `WWW-Authenticate` header, aligning with RFC 7617, which specifies that a `realm` is required. This is a positive change as it ensures compliance with the standard, potentially improving interoperability and security.
- 2. Potential Information Disclosure:** If the `realm` is set to an empty string, it might not provide meaningful context to the client about the authentication scope.

Demo Link

- Please find the Demo of ReviewGenie below.

 [ReviewGenie Demo](#)

Thanks