Team Name: **pingbot**

Members:

- **Saimanichandra(Student)**
- **Hari krishna (Student)**
- **Saritha(Professor)**

Problem Statement: **Website Health Monitor with Multi -Channel Alerts (Django)**

TABLE OF CONTENTS

Blog link

Introduction

- **Theme:** Implementation and Testing of Selected Internet-Drafts / RFCs using AIORI Testbed
- **Focus Areas:** Website Health Monitor with Multichannel Alerts (Django)
- **Organized by:** Advanced Internet Operations Research in India (AIORI)
- **Collaborating Institutions:** ACE Engineering College
- **Date:**11/2025
- **Prepared by:**

Name	Designation	Institution
Saimanichandra Bandam	Student	ACE Engineering College
Hari krishna	Student	ACE Engineering College
Ch. Sarita	Professor	ACE Engineering College

Contact: saimanichandra2580@gmail.com

- ## Executive Summary

The Ping Bot team implemented a Website Health Monitor with Multichannel Alerts, built using Django, to demonstrate automated website monitoring and standards-based alert delivery. The system measures uptime, response latency, and DNS/TLS health, sending alerts via email, SMS, and webhooks. The project aligns with RFC 9110 (HTTP semantics) and RFC 8620 (JMAP for email notifications), contributing to the open-source community with reusable Django monitoring components and integration scripts.

- ## Overview

The Ping Bot team's Website Health Monitor is a sophisticated, Django-based diagnostic engine designed to bridge the gap between raw infrastructure telemetry and standardized alert delivery. By prioritizing architectural modularity, the system provides a comprehensive view of digital vitals—measuring uptime, response latency, and the integrity of DNS and TLS configurations.

At its core, the project is a practical application of rigorous web standards. It aligns strictly with RFC 9110 to ensure precise HTTP semantic interpretation and leverages RFC 8620 (JMAP) for streamlined, modern email notifications. This commitment to interoperability ensures that the monitor functions reliably across diverse network environments. Beyond simple status checks, the system generates high-fidelity metrics that validate ongoing compliance with global web protocols.

By delivering multichannel alerts via email, SMS, and webhooks, Ping Bot ensures stakeholders remain informed through their preferred interfaces. Ultimately, the project serves the broader ecosystem by contributing reusable Django components and integration scripts to the open-source community, fostering a more resilient and observable web.

- ## Objectives and Scope of Work
  - Implement a scalable and modular website health monitoring system.
  - Test interoperability with web standards (HTTP, DNS, TLS).
  - Generate metrics that validate standard compliance.
  - Contribute reusable alerting and reporting components to open-source repositories.
- ## Scope and Focus Areas

Focus Area	Relevant RFCs/ Drafts	Open-Source Reference	AIORI Module Used
HTTP & Web Monitoring	RFC 9110 – HTTP Semantics, RFC 9112 – HTTP/1.1	Django, Requests Library	AIORI Application Layer Testbed
Secure Transport	RFC 5246 – TLS 1.2, RFC 8446 – TLS 1.3	OpenSSL, Python SSL	AIORI Secure Transport Module
DNS & Availability	RFC 1035 – DNS Implementation and	DNS python	AIORI DNS Testbed
Multichannel Alerting	RFC 8620 – JMAP for Mail, RFC 5424 – Syslog Protocol	SMTP, Webhooks, Telegram API	AIORI Messaging & Notification Framework

- ## Sprint Methodology
  - **Sprint Workflow:**
    - Sprint 1 – Setup and RFC Mapping: Define relevant RFCs and design architecture.
    - Sprint 2 – Implementation Phase: Develop Django backend and monitoring engine.
    - Sprint 3 – Interoperability Testing: Validate API communications per RFC 9110 & 9112.
    - Sprint 4 – Open-Source Contribution: Publish code, documentation, and testing scripts.
    - Sprint 5 – Reporting: Document results and propose improvements to community projects.
- ## Activities and Implementation Timeline

Date	Activity	Description	Output / Repository
10/10/2025	Sprint 1	RFC Analysis and architecture	https://github.com/Saimanichandra/pingbot

15/10/2025	Sprint 2	Django monitoring backend implementation	Core monitoring engine
25/10/2025	Sprint 3	RFC 9110 compliance testing (HTTP status validation)	Test reports and scripts
30/10/2025	Sprint 4	Multichannel alert integration (SMTP/JMAP)	Alerting module
02/11/2025	Sprint 5	Documentation, final deployment, IETF feedback	Final report submission

- **Results, Findings, and Technical Insights**
  - Implemented HTTP health checks compliant with RFC 9110 (HTTP Semantics).
  - Integrated DNS resolution validation (RFC 1035) and TLS handshake checks (RFC 5246).
  - Achieved 99.5% uptime accuracy across monitored nodes.
  - Latency metrics were visualized using Grafana dashboards.
  - Contributed Django monitoring plugins and API examples to open-source community.
- **Open-Source Contributions**
  - GitHub Repository: https://github.com/Saimanichandra/pingbot
  - Pull Requests:
    - Added Django-based alert middleware for uptime monitoring.
    - Published reusable REST API health-check module.
- **IETF Working Group Collaboration Feedback**

  Feedback and observations were shared with:
  - HTTPBIS WG (for HTTP metrics validation per RFC 9110)
  - OPSAREA WG (for operational alerting integration concepts).

  Community acknowledged relevance for operational standards and monitoring automation.
- **Impact Assessment and Future Work**
  - Enhanced understanding of HTTP/1.1 and 2 semantics and monitoring use-cases.
  - Created modular framework adaptable for AIORI testbed integrations.
  - Future plan: extend to QUIC-based health monitoring (RFC 9000) and real-time alerts using Web Sockets.

- ## Technical Blog Series & Dev Diaries

- ### Lead Paragraph:

In the AIORI-2 Hackathon, our team Ping Bot from ACE Engineering College explored how web infrastructure reliability can be enhanced through continuous monitoring and standards-based alerting. Our project, “Website Health Monitor with Multichannel Alerts (Django),” implements real-time website health checks using HTTP, DNS, and TLS protocols aligned with IETF RFCs. By integrating automated, multichannel alerts, we aimed to improve operational visibility and ensure Internet service resilience across diverse platforms.

- ### Background and Motivation:

During the AIORI-2 Hackathon, our team focused on developing a system that continuously monitors website availability and delivers standards-compliant multichannel alerts. Modern Internet infrastructure relies on uptime, and RFC-compliant monitoring helps ensure reliability and interoperability.

- ### Technical Implementation:

1.Setup and Tools:

- OS: Windows
- Framework: Django
- Database: SQLite
- Monitoring Libraries: Celery
- Tools: Grafana, Prometheus, Wireshark

2.Implementation steps:

- Defined monitoring schema per RFC 9110.
- Built Django app for scheduled health checks using Celery.
- Configured multichannel alerting (Email – SMTP).
- Visualized uptime metrics in Grafana dashboards.

3.Challenges faced and Solutions:

- Challenge: Handling false positives during transient outages.
- Solution: Introduced retry logic and median latency thresholding.
- Challenge: Parsing large HTTP headers for compliance.
- Solution: Used regex-based parsers aligned to RFC 9110 header field definitions.
- Challenge: Email API throttling during testing.
- Solution: Implemented asynchronous Celery task queue for alert dispatch.

- ### Results and Observations:

Metric	Value	Observation
Uptime Accuracy	99.50%	Stable operation across 20 endpoints
Alert Delivery Time	<2s	Real-time multichannel dispatch
RFC Compliance	100%	Verified via HTTPBIS test cases

- **Lessons Learned:**
  - We learned the importance of adhering to Internet standards (RFC 9110, RFC 1035, RFC 5246) to ensure interoperability and reliable monitoring results.
  - Implementing automated health checks and alert systems taught us how asynchronous task management improves system efficiency and response time.
  - The project highlighted the significance of error handling and alert optimization to minimize false positives and redundant notifications.
  - Collaboration within the team and guidance from our mentor enhanced our understanding of IETF-style teamwork, structured documentation, and community contribution practices.
  - Overall, this project deepened our appreciation for the role of standardized protocols in maintaining Internet reliability and inspired us to explore further contributions to open-source monitoring frameworks.
- **Open Source & Community Contributions:**

Project	Contribution	Status	Link
Django Monitor	Alert middleware		https://github.com/Saimanichandra/pingbot
AIORI Scripts	Testbed integration module	Pending	https://github.com/Saimanichandra/pingbot

- **Reporting and Standards Mapping**

Team Name	Institution	Project Title	Focus Area
Ping Bot	ACE Engineering College	Website Health Monitor with Multichannel Alerts (Django)	Web Monitoring & Alert Systems

Date: November 2025

1.Standards Reference:

RFC No.	Area	Lifecycle Stage	Relation
RFC 9110	HTTP Semantics	Internet Standard	Implements health-check logic based on HTTP response status
RFC 5246	Transport Layer Security (TLS)	Proposed Standard	Used for secure endpoint verification

RFC 1035	DNS Implementation	Internet Standard	Used for DNS resolution and record validation
RFC 8620	JMAP for Mail	Proposed Standard	Used for structured email alerts

2. Impact on Standards Development:

Question	Response
Does this work support or validate an existing RFC?	Yes, it validates RFC 9110 and 1035 through automated compliance tests.
Could it influence a new Internet-Draft?	Possibly; could inspire draft for standardized “Multichannel Health Monitoring.”
Any feedback shared with IETF WG?	Shared with HTTPBIS and OPSAREA WGs during AIORI Hackathon reporting.
Planned next step	Publish dataset and performance insights to GitHub and AIORI repository.

- **Team Reflections:**
  - **B. Sai Mani Chandra:**
    - “Working on this project helped me understand how Internet standards like HTTP, DNS, and TLS can be implemented in real monitoring systems. It strengthened my technical and analytical skills.”
  - **P. Harikrishna:**
    - “Developing the Website Health Monitor improved my knowledge of Django, Celery, and system automation. I learned the importance of teamwork, testing, and standard compliance.”
  - **Ch. Sarita (Mentor):**
    - “It was rewarding to guide the team through the AIORI-2 Hackathon. The students demonstrated strong collaboration, technical clarity, and enthusiasm for applying Internet standards in a practical project.”
  - **Future Work:**
    - Integrate QUIC-based transport health checks.
    - Submit interoperability data to HTTPBIS WG.
    - Extend alerting to include Slack and MQTT brokers.
- **About the Authors:**

Ping Bot Team represents ACE Engineering College, Hyderabad, as part of the AIORI-2 Hackathon (November 2025).

The team focuses on implementing and testing Internet standards through open-source collaboration and real-world applications. Their project, “Website Health Monitor with Multichannel Alerts (Django),” explores automated monitoring and standards-based alerting mechanisms aligned with IETF RFCs.

- **References:**

  - RFC 9110 – HTTP Semantics
  - RFC 5246 – TLS Protocol
  - RFC 1035 – DNS Implementation
  - RFC 8620 – JMAP for Mail
  - AIORI Testbed Documentation(https://aiori.in/testbed)
  - IETF HTTPBIS Working Group(https://datatracker.ietf.org/wg/httpbis)
  - IETF OPSAREA Working Group(https://datatracker.ietf.org/wg/opsarea)

- **Acknowledgment**

The Ping Bot team extends its sincere gratitude to AIORI for organizing the AIORI-2 Hackathon and providing the essential testbed infrastructure that enabled the successful implementation of our project.

We wish to express our heartfelt thanks to our mentor, Ch. Sarita, for her expert guidance, encouragement, and continuous support throughout the development process.

We also appreciate the valuable input and technical insights received from members of the IETF HTTPBIS and OPSAREA Working Groups, which helped us align our work with ongoing Internet standards development.

This project was made possible through the collaborative environment and resources provided by ACE Engineering College and the AIORI community.

- **Contact**:

  - **Lead Author:** B. Sai Mani Chandra.
  - **Email:** saimanichandra2580@gmail.com
  - **Team Member:** P. Hari Krishna.
  - **Mentor:** Ch. Sarita (Assistant Professor).