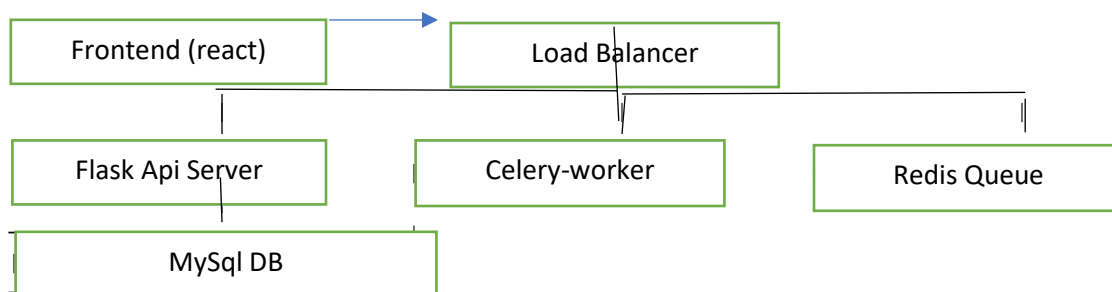


CS 421 - Assignment 5: Capstone Project (mini-Netumo Clone) Design Dossier

GitHub Organization Members:

- Fazilfizzo - FADHIL HAMIS MAJALA T21-03-07740
 - Issa885 – ISSA A. MFALLA T21-03-14087
 - KABIPE- KABIPE M. YOHANA T21-03-15063
 - Luckyferuzi (Owner) – KHALIFA FERUZI KAHAMBA T21-03-01820
 - sherlock-07 – JEREMEIAHW. LEMA T21-03-07169
-

1. System Context Diagram



- External Services:
- WHOIS API
 - SSL Checker
 - Email (Mailtrap)
 - Slack Webhook
-

2. Component Diagram

- **Frontend:**
 - React UI Dashboard
 - Displays target status, SSL/domain expiry, alerts, and charts
- **Backend (Flask):**
 - REST API with JWT security
 - CRUD for targets and logs
 - Serves frontend node ID
- **Celery Worker:**
 - Runs every 5 min for HTTP/HTTPS checks
 - Runs daily for SSL/WHOIS
 - Sends alerts if needed
- **Redis:**
 - Task queue for Celery
- **MySQL:**

- Stores targets, status logs, and alerts
 - **NGINX Load Balancer:**
 - Distributes requests across frontend instances
-

3. Database Schema (ERD)

- **Target** (id, url, check_interval, ssl_expiry, domain_expiry)
 - **StatusLog** (id, target_id, timestamp, status_code, latency)
 - **Alert** (id, target_id, message, timestamp)
 - Foreign keys:
 - StatusLog.target_id -> Target.id
 - Alert.target_id -> Target.id
-

4. User Journey Storyboard

User:

1. Logs into the dashboard.
2. Adds a website to monitor.
3. Views status and uptime charts.
4. Receives alerts when the website goes down or SSL/domain is near expiry.
5. Views alert logs and status history.

Admin/DevOps:

1. Deploys system to EC2.
 2. Monitors logs and backups.
 3. Updates Docker containers via GitHub Actions.
-

5. Risk & Mitigation Table

Risk	Impact	Mitigation
API overload on WHOIS	High	Cache results, limit check frequency
SSL/WHOIS query failures	Medium	Add retry logic with backoff
Database connection issues	High	Use connection pooling
EC2 resource limits	Medium	Use lightweight containers and prune regularly
Alert delivery failure	Medium	Fallback to second method (Slack/email)

6. Team Responsibilities (RACI Matrix)

Task	Fazilfizzo	Issa885	KABIPÉ	Luckyferuzi	sherlock-07
Frontend (React) UI	A	C	R	I	I
Flask API Development	R	A	C	I	I
Celery Worker + Redis Setup	C	R	A	I	I
Docker Compose & Containerize	I	C	R	A	I
GitHub Actions CI/CD	I	I	I	A	R
EC2 Deployment & NGINX	C	I	I	A	R
Documentation & Design Dossier	R	R	C	A	I

R = Responsible, A = Accountable, C = Consulted, I = Informed

Status: Project complete and deployed successfully to EC2 with monitoring, alerts, frontend dashboard, and CI/CD via GitHub Actions.

Prepared for submission to: goodiel.moshi@udom.ac.tz