

Perfect! Here’s a **full package** for you: the **complete report** combined with a **diagram layout** ready for Canva or documentation purposes.

Teeth-Management-System Deployment Report

Prepared for: Joseph George Wahba
Date: 28-11-2025

1 Project Overview

- **Frontend:** React (Vite)
 - **Backend:** Flask APIs (chatbot: port 5010, another API: port 8080)
 - **Web Server:** Apache 2.4 on Ubuntu (HTTPS via SSL)
 - **Domain:** thoutha.page
 - **Goal:** Serve React frontend, configure reverse proxy for APIs, enable SSL, and run Flask in production mode with logging.
-

2 Frontend Setup

Steps Completed:

1. Built frontend using Vite:
`npm run build`
2. Served the dist/ folder with Apache:
 - DocumentRoot: /var/www/html
 - Enabled SSL with .crt and .key files from UbiquiTLS.

Errors & Solutions:

Error	Cause	Solution
Forbidden: You don't have permission to access this resource	Directory permissions not allowing Apache	Adjusted <Directory> block: Require all granted
Apache failed to restart: AH00526: Syntax error	Wrong SSL file path	Corrected SSLCertificateFile and SSLCertificateKeyFile paths
White page on port 5173	React dev server running; Apache not serving build folder	Verified Vite build and Apache serving dist/ folder

Error	Cause	Solution
Vite dev server blocked by allowedHosts	Host not allowed in Vite config	Added domain to vite.config.js if needed for dev; in production use build folder

3 Reverse Proxy Setup

Goal: Forward API requests through Apache to local Flask servers:

```
ProxyPreserveHost On
ProxyPass "/api/chat" "http://127.0.0.1:5010/"
ProxyPassReverse "/api/chat" "http://127.0.0.1:5010/"

ProxyPass "/api/other" "http://127.0.0.1:8080/"
ProxyPassReverse "/api/other" "http://127.0.0.1:8080/"
```

Errors & Solutions:

Error	Cause	Solution
Not Found for /api/chat	Backend route mismatch	Adjusted Apache to forward /api/chat to the correct Flask endpoint /chat
Postman ENOTFOUND thoutha.page	Typo in domain	Corrected to thoutha.page

Notes: Frontend fetch requests must match proxy paths. SPA routing handled via .htaccess fallback to index.html.

4 Flask Backend Deployment (Production Mode)

Steps Taken:

1. Installed Gunicorn:

```
pip3 install gunicorn
```

2. Tested Gunicorn manually:

```
gunicorn --bind 127.0.0.1:5010 api:app
```

3. Created systemd service: /etc/systemd/system/flask-api.service

Errors & Fixes:

Error	Cause	Solution
Failed to locate executable /usr/local/bin/gunicorn (status=203/EXEC)	Wrong path to Gunicorn	Used which gunicorn to find correct path (e.g., /home/ubuntu/.local/bin/gunicorn)
Service failed to start multiple times	Path issue + permissions	Updated ExecStart to correct Gunicorn path, ensured user ubuntu has access

Production systemd service example:

```
[Unit]
Description=Flask Production API
After=network.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/Teeth-Management-System/Flask-API
ExecStart=/home/ubuntu/.local/bin/gunicorn --bind 127.0.0.1:5010 app:app \
    --access-logfile /home/ubuntu/Teeth-Management-System/logs/process_logs/ai_chatbot_api.log \
    --error-logfile /home/ubuntu/Teeth-Management-System/logs/process_logs/ai_chatbot_api.log \
    --log-level info
Restart=always

[Install]
WantedBy=multi-user.target
```

5 Logging

Existing Logging Structure:

```
logs/
├── astart_activity.log
├── process_logs/
│   ├── ai_chatbot_api.log
│   ├── ai_chatbot_api_<timestamp>.log
│   └── other API logs
```

Setup:

- Gunicorn logs configured via --access-logfile and --error-logfile pointing to process_logs/ai_chatbot_api.log

- Optional rotation with logrotate:

```
/home/ubuntu/Teeth-Management-System/logs/process_logs/ai_chatbot_api.log {  
    daily  
    rotate 14  
    compress  
    delaycompress  
    missingok  
    notifempty  
    copytruncate  
}
```

6 Final Deployment Notes

- Apache serves **React frontend** at `https://thoutha.page`.
- APIs are proxied:
 - `/api/chat` → Flask 5010
 - `/api/other` → Flask 8080
- SSL verified and active.
- Flask runs via **Gunicorn** managed by **systemd**.
- Logging handled inside `process_logs`.

7 Remaining / Optional Improvements

1. Add automated log rotation timestamps.
2. Restrict firewall to ports 80/443.
3. Monitor Gunicorn using `systemd`.
4. Optionally migrate to Docker or place NGINX in front for scaling.

Deployment Diagram

