AIML SRE Site – First Load, React/Vite Flow & API Workflow

This document explains, in human-friendly terms, what happens when a user opens https://aimlsre.com for the first time, how React + Vite initialize, and exactly which files in your repository are involved when a new button like **AI News** is added and wired end-to-end.

1. First Browser Request (Cold Visit)

User enters **https://aimlsre.com** in the browser.
CloudFront receives the request and routes it to the **S3 origin** (because this is NOT /api/*).

Files involved:
- Served file (runtime): ui/dist/index.html
- Source file (repo): ui/index.html

2. Static Assets Load (Vite Output)

index.html references hashed JS/CSS bundles created by Vite.
The browser downloads them from CloudFront → S3.

Runtime paths:
- /assets/index-.js
- /assets/index-.css

Source paths:
- ui/src/main.jsx (entry point)
- ui/src/App.jsx (root component)
- ui/src/pages/*
- ui/src/components/*

3. React Bootstraps (Vite Runtime)

Vite executes ui/src/main.jsx → mounts React into #root.
App.jsx loads layout, routing, and the default landing page.

No API calls happen yet unless a component explicitly triggers one (for example, a useEffect hook).

4. User Clicks a Button (Example: AI News)

The AI News button lives in the UI layer.
On click, React calls an API helper method.

Files involved:
- ui/src/pages/... (button handler)
- ui/src/api/client.js (API_BASE + axios/fetch wrapper)

5. CloudFront Routing for /api/*

The browser calls: https://aimlsre.com/api/news/latest
CloudFront sees /api/* and routes the request to API Gateway (NOT to S3).

Infra files:
- infra/modules/site/* (CloudFront behaviors)
- infra/envs/prod/main.tf
- infra/envs/prod/terraform.tfvars

6. API Gateway → Lambda Container

API Gateway invokes the Lambda function using the container image.
Lambda entry point is app.handler → lambda_handler.

Files involved:
- services/agent_api/app.py (primary handler)
- infra/modules/agent_api/* (Lambda + API GW wiring)

7. Path Normalization & Routing Logic

CloudFront forwards /api/news/latest.
Lambda normalizes the path by stripping '/api'.
Effective route becomes: /news/latest.

Therefore the handler MUST match '/news/latest', not '/api/news/latest'.

Critical function:
- _get_path(event) in services/agent_api/app.py

8. AI News Logic Execution

The handler fetches RSS + Hacker News (Algolia), filters noise, deduplicates, caches results in-memory, and returns JSON.

Code location:
- _handle_get_news_latest() in services/agent_api/app.py

9. Response Back to Browser

Lambda returns JSON → API Gateway → CloudFront → Browser.
React receives JSON and renders the News cards.

If CloudFront ever returns HTML here, it means the request was accidentally routed to the S3 origin.

10. Why First Click Sometimes Fails

On the very first request, CloudFront or Lambda cold-start timing can briefly cause S3 fallback or cache-miss behavior.
Once the API path is warmed and cached correctly, subsequent clicks work normally.

This is expected behavior in serverless + CDN systems and is now resolved for your setup.

11. Adding Any New Button (Repeatable Workflow)

1) UI: add button + handler in ui/src/pages or ui/src/components
2) UI: call API via ui/src/api/client.js

3) API: add route handler in services/agent_api/app.py
4) Rebuild & push Lambda container image (immutable tag / GIT_SHA)
5) Update infra/envs/prod/terraform.tfvars (lambda_image_uri)
6) terraform apply
7) Validate via curl and browser