

Full Stack Developer Assignment – Technical Response

Candidate: Sundar

Date: 24 October 2025

Problem 1 – Website Platform (Architecture & Design)

Overview: A scalable exhibition platform enabling microsite replication, unique branding, responsive UI, and real-time synchronization.

Recommended Tech Stack:

- 1 Frontend: React (Next.js) + Tailwind CSS for fast, responsive UI and SSR.
- 2 Backend: Node.js (NestJS/Express) for microservice architecture and APIs.
- 3 Database: PostgreSQL (primary) with Redis for caching and sessions.
- 4 Queue: RabbitMQ or AWS SQS for async task processing (email, badges).
- 5 Storage: AWS S3 + CloudFront for static files and assets.
- 6 CMS: Headless CMS (Strapi/Directus) for non-technical content updates.
- 7 Monitoring: Prometheus + Grafana; Sentry for error logging.

Scalability & Reliability:

- 1 Horizontal scaling using Dockerized microservices (ECS/Fargate).
- 2 Load balancing with AWS ALB or Cloud Load Balancer.
- 3 Caching with Redis and CDN for faster static delivery.
- 4 Async registration queue for handling spikes efficiently.
- 5 Multi-tenant schema for exhibition-level isolation and scalability.

Registration & Real-time Flow:

Frontend → API Gateway → Database → Queue → Worker → Email/QR Badge.

CMS & Theming:

Headless CMS with RBAC, reusable components, and JSON-based theming (colors, fonts, logo).

Problem 2 – Centralized Data Management (Visitors & Exhibitors)

Goal: To design a unified, secure, and efficient data system integrating both live and offline sources for analytics.

- 1 Central Database: PostgreSQL (OLTP) + Redshift/Snowflake (analytics).
- 2 Ingestion: Real-time APIs and Excel uploads through event-driven queues.
- 3 ETL: Validation, cleaning, deduplication via Airflow/Step Functions.
- 4 Notifications: SendGrid (Email), Twilio (SMS/WhatsApp) via abstraction layer.
- 5 Backup: Daily snapshots, PITR, and cross-region replication for DR.
- 6 Monitoring: Prometheus metrics, slow query logs, and queue depth alerts.

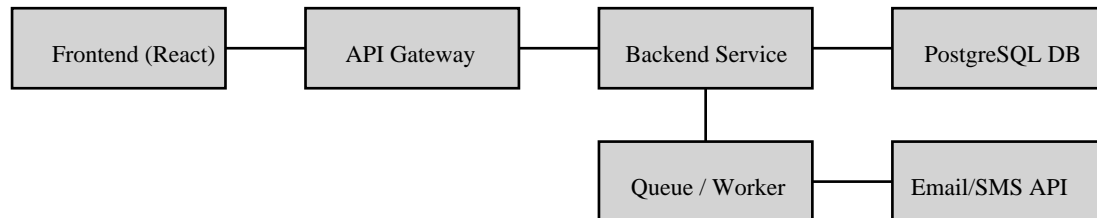
Dashboard & Access Control:

React dashboard for managing campaigns, filters, and exports with RBAC and audit logs.

Implementation Phases:

- 1 Phase 1 (2–4 weeks): Core APIs, registration queue, and email service.
- 2 Phase 2 (4–6 weeks): CMS integration, theming, and Excel/ETL pipeline.
- 3 Phase 3 (2–3 weeks): Analytics, autoscaling, and DR strategy setup.

System Architecture Overview:



Conclusion:

The proposed architecture ensures scalability, modularity, and reliability while providing non-technical teams with easy content management and secure data handling. It supports future growth, rapid replication, and robust analytics.