THEORETICAL TASKS

9. Website vs Web App; CSR vs SSR

- **Website**: Primarily static, focused on delivering information (e.g., blogs, portfolios, company pages). Limited interactivity.
- **Web App**: Dynamic, interactive, allows users to perform actions (e.g., Gmail, Trello). Requires backend logic and databases.
- **Client-Side Rendering (CSR)**:
    - Browser builds the page using JavaScript.
    - Pros : Smooth navigation after first load, rich interactivity.
    - Cons : Slower initial load, SEO challenges.
    - Example: React Single Page Applications.
- **Server-Side Rendering (SSR)**:
    - Server sends fully rendered HTML to the browser.
    - Pros : Faster first load, better SEO.
    - Cons :More server load, slower page reloads.
    - Example: Next.js, traditional PHP apps.

10. What Makes a Website Scalable

- Architecture: Modular design or microservices to handle growth.
- Load Balancing: Distributes traffic across servers to avoid overload.
- Caching: Stores frequently used data to reduce repeated queries.
- Database Optimization: Indexing, replication, sharding for efficiency.
- Cloud Infrastructure: Auto-scaling resources (AWS, Azure, GCP).
- CDN (Content Delivery Network): Delivers static assets closer to users worldwide.
- Monitoring: Tools to track performance and detect bottlenecks early.

11. Why Performance Matters (Core Web Vitals Basics)

- User Experience: Faster sites keep users engaged; slow sites increase bounce rates.
- SEO Ranking: Google uses performance metrics in search ranking.
- Business Impact: Even small delays reduce conversions and sales.
- Core Web Vitals: LCP (Largest Contentful Paint) → Measures loading speed (target < 2.5s).

- FID (First Input Delay) → Measures interactivity (target < 100ms).CLS (Cumulative Layout Shift) → Measures visual stability (target < 0.1).