

Teacher Information, Course Description and  
Lecture-wise Plan

of

Advanced Web Technologies (CSC337)

under

Pilot Project:

Uniform Teaching, Assessment and Quality  
Assurance Across All The Campuses

by

Faculty of Information Science and Technology  
COMSATS University Islamabad

## A: Teacher Information

|                                        |                                          |
|----------------------------------------|------------------------------------------|
| Course Name: Advanced Web Technologies | Credit Hours: 2+1                        |
| Course Code: CSC337                    | Pre-requisite: CSC336 - Web Technologies |
| Teacher Name:                          | Teacher E-mail:                          |
| Program:                               | Semester: Fall 2026                      |
| Department: Computer Science           | Campus:                                  |

## B: Course Contents

This course provides hands-on learning of current web technologies and production-grade full-stack engineering. Students cover enterprise architecture, REST APIs, security, authentication, OAuth 2.0, OWASP Top 10, and communication approaches including REST, GraphQL, JSON-RPC, and WebSockets. The course emphasizes modern frontend development with Next.js, including routing, rendering, server-side capabilities, performance, caching, scalability, database optimization, testing, and production deployment. Students learn to design, build, secure, optimize, and deploy modern web applications.

### Recommended Books:

1. Web Development with Node and Express, Ethan Brown, O'Reilly.
2. Node.js Design Patterns, Mario Casciaro & Luciano Mammino.
3. Designing Data-Intensive Applications, Martin Kleppmann.
4. Real-World Next.js, Michele Riva.
5. MongoDB: The Definitive Guide, Shannon Bradshaw, Eoin Brazil & Kristina Chodorow.
6. Official Next.js Documentation ([nextjs.org/learn](https://nextjs.org/learn)).

## C: Lecture-wise Plan

| Week # | Lecture #   | Topic (Chapter/Topics Covered)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Practice/<br>Evaluation                        | Remarks                           |
|--------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|-----------------------------------|
| 1      | 1<br>(Date) | <b>Course Overview and Enterprise Web Application Architecture:</b> <ul style="list-style-type: none"> <li>Course introduction, CLOs and assessment plan; how AWT builds on Web Technologies</li> <li>Web application architectures: monolithic, modular monolith, microservices and serverless</li> <li>Layered architecture: presentation, application/service, business/domain, data access and infrastructure</li> <li>Scalability and reliability fundamentals: scaling, stateless applications, load balancing, single points of failure</li> </ul>                                                                                                                                                                         | Exercises as given in the books                | Casciaro & Mammino, Chapter No. 1 |
|        | 2<br>(Date) | <b>Technology Selection and Cross-Cutting Concerns:</b> <ul style="list-style-type: none"> <li>Microservices fundamentals: service boundaries, benefits, challenges and inter-service communication</li> <li>Architecture and technology selection: when to choose what.</li> <li>Trade-offs: complexity, cost, performance and maintainability</li> <li>Cross-cutting concerns: security, performance (latency, throughput, caching), communication and deployment</li> <li>Multi-tenancy SaaS Architecture</li> </ul>                                                                                                                                                                                                           | Exercises as given in the books                | Casciaro & Mammino, Chapter No. 1 |
| 2      | 3<br>(Date) | <b>Full-Stack Application Architecture - Business, Infrastructure and Application Layers</b> <ul style="list-style-type: none"> <li>What is software architecture; the layers of a full-stack application: presentation, business, infrastructure, application and data</li> <li>Business layer: domain understanding, requirements, business rules, use cases, user stories; functional vs. non-functional requirements</li> <li>Infrastructure layer: servers, VMs, containers, orchestration, cloud service models (IaaS/PaaS/SaaS), load balancing and CDN</li> <li>Application layer: API gateway, service layer, cross-cutting concerns, stateful vs. stateless design, fault tolerance and graceful degradation</li> </ul> | Exercises as given in the books + Assignment 1 | Kleppman, Chapter No. 1 & 2       |
|        | 4<br>(Date) | <b>Data Layer and Microservices Architecture:</b> <ul style="list-style-type: none"> <li>Data layer design: data modelling, SQL vs. NoSQL, polyglot persistence,</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Quiz # 1                                       | Kleppman, Chapter                 |

|   |             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                 |                                                     |
|---|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|-----------------------------------------------------|
|   |             | <p><i>replication/partitioning, indexing and the CAP theorem</i></p> <ul style="list-style-type: none"> <li>• <i>Monolithic vs. modular-monolith vs. microservices vs. serverless architectures selection</i></li> <li>• <i>Microservices principles: service decomposition, bounded contexts and service boundaries</i></li> <li>• <i>Inter-service communication: REST, gRPC, message queues; service discovery, API gateway, database-per-service</i></li> <li>• <i>Benefits, challenges and anti-patterns of microservices; when a monolith is the better choice; architecture/technology selection trade-offs</i></li> </ul>                                                                                                                              |                                 | No. 1 & 2                                           |
| 3 | 5<br>(Date) | <p><b>Professional API Design Practices:</b></p> <ul style="list-style-type: none"> <li>• <i>REST architectural constraints: client-server, statelessness, cacheability, layered system, uniform interface</i></li> <li>• <i>Resource modelling and URI naming conventions; nouns vs. verbs, nesting of resources; HTTP methods, idempotency and safe methods</i></li> <li>• <i>Status codes and consistent error responses; pagination, filtering, sorting and partial responses</i></li> <li>• <i>API versioning strategies and HATEOAS; API documentation with OpenAPI/Swagger and API-Gateway basics</i></li> <li>• <i>Breaking vs. non-breaking changes</i></li> <li>• <i>Deprecation policy, sunset headers and consumer-driven contracts</i></li> </ul> | Exercises as given in the books | Ethan Brown<br>Chapter No 15 -18                    |
|   | 6<br>(Date) | <p><b>REST and GraphQL</b></p> <ul style="list-style-type: none"> <li>• <i>REST recap: resource-oriented request/response; over-fetching and under-fetching</i></li> <li>• <i>GraphQL schema, types and the Schema Definition Language</i></li> <li>• <i>Queries, mutations, subscriptions and resolvers</i></li> <li>• <i>When GraphQL makes sense; caching and query-complexity concerns</i></li> </ul>                                                                                                                                                                                                                                                                                                                                                      | Exercises as given in the books | GraphQL official documentation                      |
| 4 | 7<br>(Date) | <p><b>JSON-RPC and WebSockets</b></p> <ul style="list-style-type: none"> <li>• <i>JSON-RPC: procedure-oriented communication, message format, batching and notifications</i></li> <li>• <i>WebSocket protocol and handshake; persistent bidirectional communication</i></li> <li>• <i>Real-time communication with Socket.IO: server/client setup, events, rooms, namespaces, broadcasting; scaling with the Redis adapter</i></li> <li>• <i>Comparison with Server-Sent Events and long</i></li> </ul>                                                                                                                                                                                                                                                        | Exercises as given in the books | JSON-RPC 2.0 specification; Socket.IO documentation |

|   |              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                        |                                        |
|---|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|----------------------------------------|
|   |              | <i>polling; real-time use cases (chat, notifications, dashboards)</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                        |                                        |
|   | 8            | <b>Selecting a Suitable Communication Approach</b> <ul style="list-style-type: none"> <li>• Requirement-driven technology-selection criteria</li> <li>• CRUD/resource API → REST; flexible client queries → GraphQL</li> <li>• Procedure-oriented API → JSON-RPC; real-time bidirectional → WebSockets</li> <li>• Non-functional factors (caching, tooling, security, expertise); decision exercise on a case study</li> </ul>                                                                                                                                                                                                           | <i>Exercises as given in the books</i> | <i>Kleppman, Chapter No. 4</i>         |
| 5 | 9<br>(Date)  | <b>Application Security: Authentication and Authorization</b> <ul style="list-style-type: none"> <li>• Authentication vs. authorization; stateful sessions vs. stateless tokens</li> <li>• Session-based authentication and server-side session stores</li> <li>• Token-based authentication and JWT: structure, signing, expiry and refresh tokens; secure token storage (httpOnly cookies vs. localStorage)</li> <li>• Role-Based Access Control (RBAC); overview of access-control models (DAC, MAC, RBAC, ABAC)</li> <li>• Password hashing (bcrypt) and salting, password policies, multi-factor authentication overview</li> </ul> | <i>Exercises as given in the books</i> | <i>Ethan Brown Chapter No 13</i>       |
|   | 10<br>(Date) | <b>OAuth 2.0 and OpenID Connect</b> <ul style="list-style-type: none"> <li>• OAuth 2.0 as an authorization framework: roles and terminology</li> <li>• Authorization Code flow with PKCE; overview of other grant types</li> <li>• Access tokens, refresh tokens, scopes and user consent</li> <li>• OpenID Connect and the ID token; social login and third-party identity providers</li> <li>• Passport.js strategies; common OAuth mistakes and secure implementation</li> </ul>                                                                                                                                                      | <i>Exercises as given in the books</i> | <i>OAuth 2.0 / OIDC specifications</i> |
| 6 | 11<br>(Date) | <b>Web-Specific Security</b> <ul style="list-style-type: none"> <li>• Cross-Site Scripting (stored, reflected, DOM-based) and output encoding</li> <li>• Cross-Site Request Forgery, anti-CSRF tokens and SameSite cookies</li> <li>• CORS: preflight requests, headers and safe configuration</li> </ul>                                                                                                                                                                                                                                                                                                                                | <i>Exercises as given in the books</i> | <i>OWASP Cheat Sheet Series</i>        |

|   |              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                          |                                   |
|---|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|-----------------------------------|
|   |              | <ul style="list-style-type: none"> <li>• <i>SQL/NoSQL injection and sanitization; HTTPS/TLS, secure cookies and Content Security Policy</i></li> </ul>                                                                                                                                                                                                                                                                                                                         |                                                          |                                   |
|   | 12<br>(Date) | <b>API Security</b> <ul style="list-style-type: none"> <li>• <i>Input validation and schema-based request validation (express-validator / Joi)</i></li> <li>• <i>Rate limiting, throttling and quotas; HPP, mongo-sanitize, xss-clean</i></li> <li>• <i>API keys vs. tokens and mutual TLS; secrets management and key rotation</i></li> <li>• <i>Authorization at the API and resource level; object-level access checks</i></li> </ul>                                       | <i>Exercises as given in the books + Assignment No 2</i> | <i>OWASP API Security Top 10</i>  |
| 7 | 13<br>(Date) | <b>OWASP Top 10</b> <ul style="list-style-type: none"> <li>• <i>Purpose, structure and correct use of the OWASP Top 10</i></li> <li>• <i>Broken access control, cryptographic failures and injection</i></li> <li>• <i>Insecure design, security misconfiguration and vulnerable/outdated components</i></li> <li>• <i>Authentication and integrity failures, logging/monitoring failures, SSRF; practical hardening checklist (Helmet, disabling x-powered-by)</i></li> </ul> | <i>Exercises as given in the books</i>                   | <i>OWASP Top 10 (2021)</i>        |
|   | 14<br>(Date) | <b>Frontend Performance</b> <ul style="list-style-type: none"> <li>• <i>Core Web Vitals (LCP, CLS, INP) and measurement with Lighthouse</i></li> <li>• <i>HTTP and browser caching of static assets</i></li> <li>• <i>Asset optimisation: images, fonts, compression and minification</i></li> <li>• <i>Lazy loading, code splitting and bundle analysis</i></li> </ul>                                                                                                        | <b>Quiz # 2</b>                                          | <i>web.dev performance guides</i> |
| 8 | 15<br>(Date) | <b>Web / Server Caching</b> <ul style="list-style-type: none"> <li>• <i>Cache-Control, ETag, Last-Modified and cache validation</i></li> <li>• <i>Private browser cache vs. shared caches</i></li> <li>• <i>CDN and edge caching; reverse-proxy caching with Nginx/Varnish</i></li> <li>• <i>Cache keys, invalidation and purging strategies</i></li> </ul>                                                                                                                    | <i>Exercises as given in the books</i>                   | <i>Kleppman, Chapter No. 1</i>    |
|   | 16           | <b>Application Caching with Redis</b>                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                          |                                   |

|    |              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                          |                                                               |
|----|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|---------------------------------------------------------------|
|    |              | <ul style="list-style-type: none"> <li>• <i>Introduction to Redis: in-memory data store and data structures (strings, hashes, lists, sets, sorted sets)</i></li> <li>• <i>Cache-aside, read-through and write-through/write-behind patterns</i></li> <li>• <i>Cache invalidation, TTL, eviction policies and stampede protection</i></li> <li>• <i>Using Redis for API caching, session storage, rate limiting and leaderboards</i></li> </ul>                                                                                                                                                                                                                                                                   | <i>Exercises as given in the books</i>                   | <i>Redis documentation</i>                                    |
| 9  | 17<br>(Date) | <i>Midterm Examination</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | <b>Midterm (25%)</b>                                     |                                                               |
|    | 18<br>(Date) | <b>Database Monitoring, Optimization and Serverless Scalability</b> <ul style="list-style-type: none"> <li>• <i>Indexing strategies and reading query plans (EXPLAIN / ANALYZE)</i></li> <li>• <i>Query optimisation, the N+1 problem, connection pooling and slow-query monitoring</i></li> <li>• <i>Serverless functions: execution model, cold starts, statelessness and limits</i></li> <li>• <i>Horizontal vs. vertical scaling, load balancing and bottleneck identification</i></li> </ul>                                                                                                                                                                                                                | <i>Exercises as given in the books</i>                   | <i>Kleppman, Chapter No. 3-6</i>                              |
| 10 | 19<br>(Date) | <b>Event-Driven Scalability with Kafka and Multi-Tenancy SaaS Products</b> <ul style="list-style-type: none"> <li>• <i>Event-driven architecture: events, commands, producers and consumers; Kafka concepts (topics, partitions, offsets, brokers, replication)</i></li> <li>• <i>Delivery semantics (at-most-once / at-least-once / exactly-once), ordering and retention; integrating Kafka with Node.js (KafkaJS)</i></li> <li>• <i>Software as a Service: single-tenant vs. multi-tenant architectures and business model overview</i></li> <li>• <i>Tenancy models (shared DB with tenant ID, schema-per-tenant, DB-per-tenant), tenant routing, data isolation, subscription/billing basics</i></li> </ul> | <i>Exercises as given in the books + Assignment No 3</i> | <i>Kleppman, Chapter No. 11 ; Redis / Kafka documentation</i> |
|    | 20<br>(Date) | <b>Next.js Architecture and Project Setup</b> <ul style="list-style-type: none"> <li>• <i>Recap of React fundamentals: components, props, state and hooks; limitations of client-side-only React</i></li> <li>• <i>Next.js features: hybrid rendering, file-system routing, built-in optimization and full-stack capability</i></li> <li>• <i>Rendering strategies overview: CSR, SSR, SSG</i></li> </ul>                                                                                                                                                                                                                                                                                                        | <b>Quiz # 3</b>                                          | <i>Riva, Chapter No. 1-2</i>                                  |

|    |              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                        |                                    |
|----|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|------------------------------------|
|    |              | <i>and ISR - concepts and trade-offs</i> <ul style="list-style-type: none"> <li>• <i>Project setup (create-next-app), folder structure, configuration files; App Router vs. Pages Router</i></li> </ul>                                                                                                                                                                                                                                                                                                                                                       |                                        |                                    |
| 11 | 21<br>(Date) | <b>Routing and Layout Management</b> <ul style="list-style-type: none"> <li>• <i>File-based routing conventions and special files (page, layout, loading, error, not-found)</i></li> <li>• <i>Nested routes, route groups, dynamic, catch-all and optional catch-all segments</i></li> <li>• <i>Layouts, templates and shared UI</i></li> <li>• <i>Navigation: Link component, useRouter/usePathname, programmatic navigation, redirects and rewrites</i></li> </ul>                                                                                          | <i>Exercises as given in the books</i> | <i>Riva, Chapter No. 3</i>         |
|    | 22<br>(Date) | <b>Rendering Strategies and Server-Side Capabilities</b> <ul style="list-style-type: none"> <li>• <i>Server Components vs. Client Components; the “use client” directive and composition rules</i></li> <li>• <i>Server-side data fetching, request memoization, caching and revalidation (time-based and on-demand ISR)</i></li> <li>• <i>Streaming and Suspense; loading skeletons and progressive rendering</i></li> <li>• <i>Route handlers / API routes, Server Actions and Next.js middleware</i></li> </ul>                                            | <i>Exercises as given in the books</i> | <i>Riva, Chapter No. 4</i>         |
| 12 | 23<br>(Date) | <b>Components, Layouts and Styling</b> <ul style="list-style-type: none"> <li>• <i>Special-file behaviour; Metadata API for titles, descriptions and SEO; built-in optimizations (next/image, next/font, next/script)</i></li> <li>• <i>Styling options: global CSS, CSS Modules, Tailwind CSS and CSS-in-JS</i></li> <li>• <i>Component libraries (shadcn/ui, MUI) and design systems; responsive design, dark mode and accessibility</i></li> <li>• <i>Building the application shell: navigation, sidebar, forms and reusable UI components</i></li> </ul> | <i>Exercises as given in the books</i> | <i>Riva, Chapter No. 6 &amp; 7</i> |
|    | 24<br>(Date) | <b>Frontend Architecture</b> <ul style="list-style-type: none"> <li>• <i>Component organisation and folder conventions at scale</i></li> <li>• <i>State categories: local UI state, server state and URL state</i></li> <li>• <i>API/service layer and data-access abstraction; form handling and schema validation</i></li> <li>• <i>Error boundaries, loading skeletons and code splitting</i></li> </ul>                                                                                                                                                   | <i>Exercises as given in the books</i> | <i>Riva, Chapter No. 6 &amp; 7</i> |
| 13 | 25<br>(Date) | <b>Data Fetching and State Management with TanStack Query:</b> <ul style="list-style-type: none"> <li>• <i>Client-side data fetching with SWR / TanStack</i></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                       | <i>Exercises</i>                       | <i>TanStack</i>                    |



|    |              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                |                              |
|----|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|------------------------------|
|    |              | <i>Query: caching, staleness, mutations and optimistic updates</i> <ul style="list-style-type: none"> <li>• <i>Query keys, pagination, infinite queries, prefetching, retries and error handling</i></li> <li>• <i>State-management options in Next.js (Context, Zustand, Redux Toolkit) and when to use each</i></li> <li>• <i>Handling errors, empty states and performance pitfalls in data fetching</i></li> </ul>                                                                                                                                                          | <i>as given in the books + Assignment No 4</i> | <i>Query documentation</i>   |
|    | 26<br>(Date) | <b>Authentication and Security in Next.js</b> <ul style="list-style-type: none"> <li>• <i>Authentication approaches in Next.js: session-based, JWT and third-party providers</i></li> <li>• <i>NextAuth.js / Auth.js: setup, providers, callbacks, sessions and adapters</i></li> <li>• <i>Protecting pages, layouts, route handlers and server actions; middleware-based route protection and role-based UI rendering</i></li> <li>• <i>Secure handling of environment variables (server-only secrets vs. public variables); XSS, CSRF, CORS and secure headers</i></li> </ul> | <b>Quiz # 4</b>                                | <i>Riva, Chapter No. 4</i>   |
| 14 | 27<br>(Date) | <b>Performance Optimization and Testing in Next.js</b> <ul style="list-style-type: none"> <li>• <i>Performance metrics and Core Web Vitals (LCP, CLS, INP)</i></li> <li>• <i>Code splitting, dynamic imports, lazy loading and bundle analysis; image/font/asset optimization</i></li> <li>• <i>SEO in Next.js: metadata, sitemap, robots.txt, structured data and Open Graph</i></li> <li>• <i>Testing the application: unit, component and end-to-end testing (Jest, React Testing Library, Playwright/Cypress)</i></li> </ul>                                                | <i>Exercises as given in the books</i>         | <i>Riva, Chapter No. 9</i>   |
|    | 28<br>(Date) | <b>Deploying the Product on Vercel</b> <ul style="list-style-type: none"> <li>• <i>Build process, production build output and self-hosting vs. managed hosting</i></li> <li>• <i>Introduction to Vercel: projects, Git integration, preview and production deployments</i></li> <li>• <i>Configuring environment variables, secrets and build settings on Vercel</i></li> <li>• <i>Serverless and edge functions/middleware; custom domains, HTTPS, CDN/caching behaviour, analytics and CI/CD rollbacks</i></li> </ul>                                                         | <i>Exercises as given in the books</i>         | <i>Next.js documentation</i> |
| 15 | 29<br>(Date) | <b>Web Application and API Testing</b> <ul style="list-style-type: none"> <li>• <i>Testing pyramid: unit, integration and end-to-end tests</i></li> <li>• <i>Unit and integration testing tools (Jest/Vitest,</i></li> </ul>                                                                                                                                                                                                                                                                                                                                                    | <i>Exercises as given in the books</i>         | <i>Ethan Brown Chapter</i>   |

|    |              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                        |                                   |
|----|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|-----------------------------------|
|    |              | <i>React Testing Library, Supertest</i> <ul style="list-style-type: none"> <li>• <i>API testing: success paths, validation errors, authentication and edge cases; Postman collections and Swagger UI</i></li> <li>• <i>End-to-end testing (Playwright/Cypress) with mocking; coverage and CI integration</i></li> </ul>                                                                                                                                                                                                                                                                                 |                                        | No 5                              |
|    | 30<br>(Date) | <b>Reliability and Observability: Log Levels</b> <ul style="list-style-type: none"> <li>• <i>Reliability practices: retries, circuit breakers, health checks and graceful degradation; CI/CD pipeline and deployment strategies (build, test, deploy, rollback)</i></li> <li>• <i>Why logging matters: logs vs. metrics vs. traces</i></li> <li>• <i>Log levels: TRACE, DEBUG, INFO, WARN, ERROR and FATAL; structured logging with correlation/request identifiers</i></li> <li>• <i>Log aggregation, retention and handling of sensitive data; deciding what to log (signal vs. noise)</i></li> </ul> | <i>Exercises as given in the books</i> | <i>Kleppman n, Chapter No. 13</i> |
| 16 | 31<br>(Date) | <b>Application and Server Monitoring; Course Review</b> <ul style="list-style-type: none"> <li>• <i>Key metrics: latency, error rate, throughput and saturation; health checks, uptime monitoring and alerting</i></li> <li>• <i>Application performance monitoring (APM) and distributed-tracing basics</i></li> <li>• <i>Server monitoring (CPU, memory, disk, network); dashboards, SLIs/SLOs and incident response</i></li> <li>• <i>Consolidated review of Units 1-7; semester-project demonstration and final-exam guidelines</i></li> </ul>                                                      | <i>Exercises as given in the books</i> | <i>Kleppman n, Chapter No. 12</i> |
|    | 32<br>(Date) | <ul style="list-style-type: none"> <li>• <i>Review</i></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                        |                                   |
| 17 | (Date)       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | <b>Final (50%)</b>                     |                                   |

**Note:** All Assignments/Quizzes weightage is 25%