

ONE CREDIT COURSE REGISTRATION AND COURSE EXEMPTION SYSTEM

Core Backend Development (Part 2)

Aim:

To enhance the backend system by implementing additional APIs, refining business logic, and optimizing database queries to support complex use cases, ensuring efficient, scalable, and maintainable application performance.

Objectives:

1. **API Development:** Expand the backend by developing new APIs to handle advanced functionalities and integrations.
2. **Business Logic Implementation:** Ensure accurate processing of business rules and data handling to align with system requirements.
3. **Database Optimization:** Design and execute complex queries to retrieve, manipulate, and manage data efficiently.
4. **Performance Enhancement:** Improve the system's response time, scalability, and resource utilization through optimized coding practices.
5. **Security & Compliance:** Implement security best practices to protect data and ensure compliance with industry standards.
6. **Testing & Debugging:** Validate API responses, database operations, and business logic through rigorous testing and debugging.

Overview:

Core Backend Development (Part 2) focuses on extending and refining the backend infrastructure of the application. This phase includes the creation of additional APIs to accommodate new features, the development of intricate business logic to align with functional requirements, and the execution of complex database queries to support advanced data operations. The implementation aims to enhance system performance, maintainability, and scalability, ensuring seamless interaction between the frontend and backend components.

Scope:

- **API Development:**
 - Designing and implementing RESTful APIs.
 - Enhancing existing endpoints for better functionality.
 - Integrating with third-party services as required.
- **Business Logic Implementation:**
 - Incorporating rules and workflows essential for application functionality.
 - Handling edge cases and error scenarios effectively.
- **Database Management:**
 - Writing optimized SQL/NoSQL queries for data retrieval and manipulation.

- Implementing indexing, caching, and query optimization techniques.
- **Performance Optimization:**
 - Reducing API response times and optimizing database queries.
 - Load testing to ensure system stability under heavy traffic.
- **Security & Compliance:**
 - Implementing authentication and authorization mechanisms.
 - Ensuring data encryption and secure API communication.