**CHAPTER 1**

**1. INTRODUCTION**

**1.1 Background of the Capstone Project**

Describe the origins and context of the project. Provide background information such as institutional, industrial, or technological triggers.

**1.2 Context and Scope**

Define the boundaries and operational context of the project.

**1.3 Problem Statement**

Clearly outline the problem your project aims to solve.

**1.4 Objectives and Goals**

**1.5 Significance and Relevance**

Explain the importance of the project to stakeholders, organizations, or industries.

**1.6 Structure of the Document**

Briefly summarize the contents of each major chapter.

**2. REVIEW OF RELATED LITERATURE**

**2.1 Agile Scrum Methodology Overview**

Introduce Agile Scrum, including its key principles, practices, and relevance to your project.

**2.2 Enterprise Architecture Concepts**

Discuss enterprise architecture frameworks like TOGAF and their application in your project.

**2.3 Microservices Architecture**

**2.4 DevOps and CI/CD**

**2.5 Relevant Studies and Research**

Present a review of existing literature, similar projects, or technologies relevant to your capstone.

**2.6 Integration of Information Systems in Enterprise Environment**

Explain the importance, challenges, and approaches to integrating systems in enterprise settings.

**3. METHODOLOGY**

**3.1 Agile Scrum Methodology in the Project**

Describe how Agile Scrum was used to manage and execute your project.

**3.2 Roles (Scrum Master, Product Owner, Development Team)**

Define team roles and their responsibilities in your capstone project.

**3.4 Sprint Cycles (Planning, Standups, Review)**

Explain your sprint structure and how these ceremonies were implemented.

**3.5 Scrum Artifacts (Product Backlog, Sprint Backlog)**

Describe how your team managed and used these key Scrum tools.

**3.6 Microservices Architecture**

**3.6 DevOps Implementation**

**3.7 Integration Approach for Information Systems**

**3.8 Introduction to TOGAF and the Four Architectural Domains**

Introduce TOGAF and briefly explain the Business, Application, Data, and Technology Architecture domains.

**4. REQUIREMENTS ANALYSIS**

**4.1 Stakeholder Identification**

**4.2 Requirements Gathering Techniques**

Outline methods used such as interviews, surveys, document analysis, etc.

**4.3 User Stories and Use Cases**

Present your user stories (Agile format) and use case diagrams.

**4.4 Functional Requirements for Integration**

Enumerate and describe detailed integration-related system requirements.

**CHAPTER 2**

**5. BUSINESS PROCESS ARCHITECTURE**

**5.1 Identification of Business Processes**

Describe relevant business processes that the system supports or enhances.

**5.2 Business Process Diagrams**

Include visual representations of business workflows (e.g., BPA diagrams).

Include BPA or flowcharts.

**5.3 Alignment of Integrated System with Business Processes**

Discuss how the system aligns with or enhances current business processes.

Explain fit and enhancements.

**5.4 Business Process Improvements**

Highlight any process efficiency gains or innovations introduced by the project.

Compare "as-is" vs. "to-be" states.

**6. APPLICATION ARCHITECTURE**

**6.1 Components of Application Architecture**

Describe the system modules, layers, or components.

**6.2 Application Architecture Diagrams**

Include diagrams showing structure, logic, and component interaction.

**6.3 Integration of Software Modules**

Explain how internal and external modules interact.

**6.4 Communication and Interaction Patterns**

Describe technologies and protocols used (e.g., REST, SOAP, messaging queues).

**7 DATA ARCHITECTURE**

**7.1 Data Sources and Types**

Define your structured/unstructured data sources.

**7.2 Data Flow Diagrams**

Include Level 0 and Level 1 DFDs.

**7.3 Data Storage and Management**

Discuss your database model and data handling processes.

**7.4 Data Synchronization Across Systems**

Explain how consistency is maintained across systems or modules.

**8 TECHNOLOGY ARCHITECTURE**

**8.1 Technology Stack and Infrastructure**

List and describe all tools, frameworks, and infrastructure components used.

**8.2 Software Technologies**

Discuss programming languages, frameworks, and libraries.

**8.3 Scalability and Performance Considerations**

Explain how the system supports scalability and optimal performance.

**CHAPTER 3**

**9 DEVELOPMENT PROCESS**

**9.1 Agile Scrum Roles and Responsibilities**

Discuss actual role assignments during implementation.

**9.2 Sprint Planning and Backlog Management**

Describe how sprint planning was conducted and how tasks were managed.

**9.3 Sprint Execution and Deliverables**

Summarize sprint activities and highlight key outputs delivered.

**9.4 Challenges Faced in the Development Process**

Discuss difficulties encountered and how the team resolved them.

**CHAPTER 4**

**10 IMPLEMENTATIONS**

**10.1 Technical Implementation Details**

Provide a detailed explanation of how the system was deployed or installed.  
Include environments (e.g., local, staging, production) and versioning.

**10.2 Tools and Technologies Used**

Enumerate the tools, platforms, and programming languages used in implementation.

**10.3 Code Integration and Interoperability**

Explain how components were integrated and how the system ensured seamless interoperability between modules.

**10.4 Integration Testing and Debugging**

Outline how integration testing was conducted and issues that were fixed.

**11. TESTING**

**11.1 Testing Strategies and Methodologies**

Describe testing models applied (e.g., unit testing, black box, regression testing).

**11.2 Test Cases and Test Data**

Provide sample test cases and datasets used to validate system functionality.

**11.3 Test Results and Bug Reports**

Summarize actual test results and list bugs discovered and fixed.

**11.4 Quality Assurance Measures**

Describe how software quality was ensured, including coding standards, reviews, and test coverage.

**12 RESULTS AND EVALUATION**

**12.1 Project Outcomes and Deliverables**

List the final deliverables (e.g., working system, documentation, deployment scripts) and their current status.

**12.2 Alignment with Project Objectives**

Analyze how the final product meets the objectives outlined in Chapter I.

**12.3 Stakeholder and User Feedback**

Present feedback from real users or stakeholders gathered during testing or demo sessions.

**12.4 Lessons Learned**

Highlight major insights, mistakes, and improvements noted during the project lifecycle.

**CHAPTER 5**

**13 CONCLUSION AND RECOMMENDATIONS**

**5.1 Key Takeaways and Summary**

Summarize the entire project journey from planning to implementation.

**5.2 Project Achievements and Contributions**

Describe major accomplishments and how the project contributes to your field, organization, or the school.

**5.3 Future Work and Enhancements**

Recommend new features, updates, or modules that can be added post-project.

**5.4 Closing Remarks**

Provide final thoughts or messages to your mentors, institution, or future capstone groups.