

**Course Code: CSE226/CSE316/CSE317**

**Course Title: Software Project III**

**Credit: 1**

### **Course Syllabus:**

**Understanding Software Architecture Principles:** Introduction to software architecture and its significance, Key architectural concepts: modularity, scalability, maintainability, Architectural patterns: layered, client-server, microservices, etc., Role of software architecture in project success

**Translating Customer Requirements from SRS into Software Design:** Extracting design requirements from System Requirements (SRS), Design considerations for different software components, Mapping functional specifications to design elements, Creating design documents for effective communication

**Version Control using Git Platforms (GitHub, GitLab):** Importance of version control in collaborative software development, Collaborative coding workflows using Git, Branching, merging, and resolving conflicts, Utilizing Git platforms like GitHub and GitLab for project management

**Effective Communication Strategies for Team Collaboration:** Importance of clear communication in software development teams, Communication tools and best practices for remote and distributed teams, Code commenting, documentation, and knowledge sharing, Resolving conflicts and fostering collaborative work environments

**Software Design Patterns and Modularity:** Exploring common software design patterns, Principles of modularity and reusability in software design, Applying design patterns to improve software architecture, Balancing trade-offs between flexibility and complexity

**Responsiveness in Software Design:** Understanding the importance of responsive design, Techniques for designing responsive user interfaces, Implementing media queries for different devices and screen sizes, Ensuring consistent user experiences across various platforms

**Developing Frontend or GUI Components:** Principles of user interface (UI) and graphical user interface (GUI) design, Selecting appropriate technologies and frameworks for frontend development, Designing user-friendly interfaces that align with user needs, Implementing responsive and interactive frontend components, Frontend Presentation: Explaining UI implementation and design choices

**Presentation:** Students will submit and present their UI based on their prototype, Demonstrating the responsiveness of the user interface

### **Course Description / Rationale:**

This course delves deep into software development essentials, encompassing software architecture, effective design translation, frontend development, version control, and team communication. Students will master architectural concepts, design patterns, responsive design, and collaboration tools, enabling them to craft user-centered solutions while collaborating seamlessly within professional teams.

**Course Objectives:**

By the end of this course, students will be able to:

- Understand the fundamental principles of software architecture and its pivotal role in project success.
- Apply advanced techniques to translate customer requirements from SRS into well-structured software design.
- Develop user-friendly frontend or GUI components using suitable technologies and frameworks.
- Utilize version control tools and platforms proficiently to enhance collaborative software development practices.
- Communicate effectively within software development teams, employing best practices for remote and distributed collaboration.
- Apply responsive design techniques to ensure optimal user experiences across various devices and screen sizes.
- Apply software design patterns and modularity principles to enhance software architecture.

**Course Learning Outcomes (CLOs):**

CLO	Learning Outcomes
CLO 1	Understand software architecture's pivotal role in project success.
CLO 2	Translate customer requirements into effective software designs.
CLO 3	Develop user-friendly frontend components using suitable technologies.
CLO 4	Utilize version control for collaborative software development.