Skill Enhancement Courses (SEC)

Course Code	Course Name	Level	L	T	P	C	CIE	SEE	Total	Pre-requisite
2501CS17	Object Oriented Analysis & Design using UML	IC			2	2	50	50	100	ASE
2501AI34	Web Application Development using MERN Stack	AC			2	2	50	50	100	-
2501CS19	CI/CD using DevOps	AC			1	1	100	-	100	-
Total					**5**	**5**				

Object Oriented Analysis & Design using UML
(Common to CSE, IT AIML & CSE(DS))

	L	T	P	C
Course Code: 2501CS17	**0**	**0**	**2**	**2**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Explain the importance of system analysis and design in solving complex problems.
CO2: Compare object-oriented approach with traditional approach in system analysis and design.
CO3: Analyze the importance of modeling and design of various applications.
CO4: Compare various object relationships.
CO5: Show the role and function of each UML model in developing object- oriented software.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	1	-	-	3	-	-	2	2	-	-
CO2	3	1	-	-	2	-	-	2	2	-	-
CO3	3	-	2	-	2	-	-	2	2	-	-
CO4	3	2	1	-	2	-	-	2	2	-	-
CO5	2	3	-	-	2	-	-	2	2	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	-
CO2	2	-
CO3	3	-
CO4	3	-
CO5	3	-

Practice:

1. List of Case Studies:
   ATMApplication.
   Library ManagementSystem.
   Online Book Shop.
   Customer SupportSystem.
   Point ofSale.
   Familiarization with RationalRose
2. For each case study:
   2.1) Identify and analyze events
   2.2) Identify Usecases
3. For each case study:
   3.1) Develop event table
   3.2) Identify &analyze domain classes
4. For each case study:
   4.1) Represent use cases and a domain class diagram using Rational Rose

4.2) Develop CRUD matrix to represent relationships between use cases and problem domain classes

5. For each case study:
   5.1) Develop Use case diagrams
   5.2) Develop elaborate Use case descriptions &scenarios
6. For each case study
   6.1) develop system sequence diagrams
7. For each case study:
   7.1) Develop highlevel sequence diagrams for each usecase
   7.2) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing interactions among all the threelayerobjects
8. For each case study:
   8.1) Develop highlevel sequence diagrams for each usecase1
   8.2) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing interactions among all the threelayerobjects
9. For each case study:
   9.1) Develop highlevel sequence diagrams for each usecase
   9.2) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing interactions among all the threelayerobjects
10. For each case study:
    10.1) Develop highlevel Collaboration diagrams for each usecase
    10.2) Develop detailed collaboration diagrams for each use case showing interactions among all the threelayerobjects
11. For each case study:
    11.1) Develop Use case Packages.
    11.2) Develop component diagrams.
    11.3) Develop sample diagrams for other UML diagrams state chart diagrams, activity diagrams and deployment diagrams.
12. Design UML models for the following application.RestaurantSystem.

Additional practice:

Design UML models for the following Case studies.

1. Banking Application System.
2. Railway Reservation System
3. Online Movie Ticket BookingSystem.

Text Books:

1 Object-oriented analysis and design using UML, Mahesh P. Matha,PHI, ISBN: 9788120333222

2 Head first object-oriented analysis and design, Brett D. McLaughlin, Gary Pollice, Dave West,O‟Reilly. ISBN: 9780596008673

Reference Books:

1 Object-oriented analysis and design with the Unified process, John W. Satzinger, Robert B. Jackson, Stephen D. Burd, CengageLearning, ISBN: 978-0619216436

2 The Unified modeling language Reference manual, James Rumbaugh, Ivar Jacobson, Grady Booch, Addison Wesley, ISBN: 9780321245625

3 Object Oriented Analysis & Design, AtulKahate, The McGraw-Hills Companies, ISBN: 9780070583764

Web Links:

1 https://onlinecourses.nptel.ac.in/noc24_cs40/preview
2 https://www.quora.com/in/What-are-the-best-website-to-study- UML-for- beginners
3 https://creately.com/lp/uml-diagram-tool/

WEB Application Development using MERN Stack

(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501AI34	**0**	**0**	**2**	**2**

Course Outcomes:

At the end of the course, student will be able to:

- **CO1:** Make use of HTML elements, CSS styles and their attributes for designing static web pages.
- **CO2:** Apply JavaScript to develop dynamic web pages and validate forms.
- **CO3:** Build a basic web server using Node.js and also working with Node Package Manager (NPM) & Make use of Typescript to optimize JavaScript code using strict type checking
- **CO4:** Make use of router, template engine and authentication using sessions to develop application in Express JS & components, props, stats and render data in ReactJS
- **CO5:** Build a single page application using RESTful APIs in ExpressJS. & Make use of MongoDB queries

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	2	3		2			1	2		2
CO2	2	3	2		2			1	2		2
CO3	2	2	3	1	2			1	2		2
CO4	2	3	2	2	2			1	2		2
CO5	2	2	3	1	2			1	2		2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	
CO2	2	
CO3	2	
CO4	2	
CO5	2	

Practice:

1. Lists, Links, Images, Tables, Forms.
   a. Write a HTML program, to explain the working of lists.
      Note: It should have an ordered list, unordered list, nested lists and ordered list in an unordered list and definition lists.
   b. Write a HTML program, to explain the working of hyperlinks using <a> tag and href, target Attributes.
      Note: Use text to link https://www.aec.edu.in/
      Use image to link https://www.aec.edu.in/?p=Gallery
   c. Create a HTML document that has your image and your friend's image with a specific height and width. Also when clicked on the images it should navigate to their respective profiles.
   d. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan)

e. Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field.

2. HTML 5 and Cascading Style Sheets, Types of CSS & CSS Box Model
   a. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>, <footer>, <header>, <main>, <nav>, <section>, , tags.
   b. Write a HTML program, to embed audio and video into HTML web page.
   c. Write a program to demonstrate the various ways you can reference a color in CSS.
   d. Write a program, to explain the importance of CSS Box model using
      i. Content ii. Border iii. Margin iv. padding

3. Selector forms & JavaScript-internal and & I/o
   a. Write a program to apply different types of selector forms
      i. Simple selector (element, id, class, group, universal)
      ii. Combinator selector (descendant, child, adjacent sibling, general sibling)
      iii. Write a program to embed internal and external JavaScript in a web page.
      iv. Write a program to explain the different ways for displaying output.

4. Javascript Pre-defined and User-defined Objects, Conditional & Loop Statements
   a. Write a program using document, window, math & string object properties and methods.
   b. Write a program which asks the user to enter three integers, obtains the numbers from the user and outputs HTML text that displays the larger number followed by the words "LARGER NUMBER" in an information message dialog. If the numbers are equal, output HTML text as "EQUAL NUMBERS".
   c. Write a program to print 1 to 10 numbers using for, while and do-while loops.

5. Javascript Functions and Events & Node.js
   a. Design a appropriate function should be called to display
      i. Factorial of that number
      ii. ii. Fibonacci series up to that number
      iii. Write a program to show the workflow of JavaScript code executable by creating web server in Node.js.
      iv. Write a program to transfer data over http protocol using http module.
      v. Create a text file src.txt and add the following content to it. (HTML, CSS, Javascript, Typescript, MongoDB, Express.js, React.js, Node.js)
6. Typescript
   a. Write a program to understand simple and special types.
   b. Write a program to understand function parameter and return types.

7. ExpressJS – Routing, HTTP Methods,Templating
   a. Write a program to define a route, Handling Routes, Route Parameters, Query Parameters and URL building.
   b. Write a program to accept data, retrieve data and delete a specified resource using http methods.
   c. Write a program using templating engine.

8. ExpressJS – Cookies, Sessions, Authentication, Database.
    a. Write a program for session management using cookies and sessions.
    b. Write a program for user authentication.
    c. Write a program to connect MongoDB database using Mangoose and perform CRUD operations
9. ExpressJS – Templating, Form Data, Cookies, Sessions
    a. Write a program using templating engine.
    b. Write a program to work with form data.
    c. Write a program for session management using cookies and sessions.

10. ReactJS – Render HTML, JSX, Props and States
    a. Write a program to render HTML to a web page.
    b. Write a program for writing markup with JSX.
    c. Write a program to work with props and states.

11. ReactJS – Conditional Rendering, Rendering Lists, React Router
    a. Write a program for conditional rendering.
    b. Write a program for rendering lists.
    c. Write a program for routing to different pages using react router.

12. ReactJS – Hooks, Sharing data between Components, To-do list and Quiz & MongoDB – Installation, Configuration
    a. Write a program to understand the importance of using hooks.
    b. Write a program for sharing data between components.
    c. Design to-do list application.
    d. Install MongoDB and configure ATLAS

Cornerstone Project:

Project 1 : Student Management System (MERN CRUD App)

- **Implementation mainly focuses on**
    - Create, update, view, and delete student records using MongoDB, Express, React, Node.js.
- **Documentation must contain**
    - Problem Definition, Objectives, Analysis& Design, Implementation and Sample Code, Output screens, Conclusion &Enhancement

Project 2: Online Food Ordering System Using MERN

- **Implementation mainly focuses on**
    - Menu display, cart, orders, user login, admin panel.
- **Documentation must contain**
    - Problem Definition, Objectives, Analysis& Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 3: Personal Portfolio with Admin Dashboard (MERN + Authentication)

- **Implementation mainly focuses on**
  - Edit portfolio sections via a protected admin panel.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis& Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 4: To-Do List Application with User Login (MERN + Sessions/JWT)

- **Implementation mainly focuses on**
  - Add, edit, delete tasks, user registration/login, MongoDB storage.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis& Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 5: Event Registration Portal (HTML + CSS + JS + MERN Backend)

- **Implementation mainly focuses on**
  - Users register for events; admin can view/manage registrations.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis& Design, Implementation and Sample Code, Output screens ,Conclusion &Enhancement

Project 6: E-Commerce Product Catalog Using MERN

- **Implementation mainly focuses on**
  - Product listings, categories, product details, cart basics.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis & Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 7: College Information & Admission Portal (MERN Full Stack)

- **Implementation mainly focuses on**
  - Forms, tables, student admission data, admin management.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis & Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 8: Blog Posting Platform Using MERN (CRUD + File Upload)

- **Implementation mainly focuses on**
  - Users create posts, add images, edit, comment, delete.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis & Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 9: Movie Review and Rating System Using MERN

- **Implementation mainly focuses on**
  - Users can review movies, give ratings, view reviews list.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis & Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 10: Employee Record Management (MERN CRUD)

- **Implementation mainly focuses on**
  - Employee details, salary, department, search filters.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis & Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 11: Notes Application Using MERN with Authentication

- **Implementation mainly focuses on**
  - Create/edit notes, categorize them, private notes per user.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis & Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 12. Online Quiz Application Using MERN (MCQ Based)

- **Implementation mainly focuses on**
  - Admin adds questions; users take quiz; store scores.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis & Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 13: Library Book Management System

- **Implementation mainly focuses on**
  - Add books, issue/return system, student records, dashboards.
- **Documentation must contain**
  - Problem Definition, Objectives, Analysis & Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Project 14: Hospital Appointment Booking System

- **Implementation mainly focuses on**
  - Users book appointments; doctors manage schedules.
- **Documentation must contain**

Problem Definition, Objectives, Analysis & Design, Implementation and Sample Code, Output screens, Conclusion & Enhancement

Reference Books:

1 Programming the World Wide Web, Robet W Sebesta, Pearson, 7th Edition. ISBN 139789332518827

2 Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, APress, O'Reilly, 2nd edition. ISBN 9781484226520

Web Links:

1 https://infyspringboard.onwingspan.com/en/app/toc/lex_17739732834840810000_shared/overview (HTML5)

2 https://infyspringboard.onwingspan.com/en/app/toc/lex_18109698366332810000_shared/overview (Javascript)

3 https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01384329019595161635073_shared/overview (Express Js)

4 https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_013177169294712832113_shared/overview (MongoDB)

Note:

- Students must submit the Certificate of Completion offered by any Industry.
- The capstone project team size shall be four (4) students.
- Students may select any one of the above prescribed projects or a project of their own choice with the prior permission of the Course instructor.
- The SEE – Lab shall be evaluated for 50 marks based on project implementation, oral presentation, 10-minute video presentation, report and viva voce.
- The video presentation should consists of the working procedure of the project along with contribution of each student for a minimum of 2 minutes.

Finally, the Source code of the capstone Project has to be pushed into the Students GitHub repository.

CI/CD using Devops
(Common to CSE, IT, AIML & CSE (DS))

Semester:	**L**	**T**	**P**	**C**
Course Code: 2501CS19	**0**	**0**	**1**	**1**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Summarize the life cycle & principles of agile software development
CO2: Develop the fundamentals programs using Devops
CO3: Analyze the adoption of Devops in Projects and Process
CO4: Demonstrate the types of continuous integration and continuous delivery in Devops
CO5: Build an automated CICD pipeline using a stack of tools
CO6: Develop basic branching and merging in GIT

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	1	3	1	-	2	-	-	2	2	-	1
CO2	1	2	1	-	3	-	-	2	2	-	1
CO3	1	3	2	2	2	-	-	2	2	-	1
CO4	-	1	3	-	2	-	-	2	2	-	1
CO5	2	1	3	-	3	-	-	2	2	-	2
CO6	2	3	2	-	3	-	-	2	2	-	1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	1	-
CO2	1	-
CO3	2	-
CO4	2	-
CO5	3	-
CO6	2	-

Practice:

1. Get an understanding of the stages in software development lifecycle, the process models, values and principles of agility and the need for agile software development. This will enable you to work in projects following an agile approach to software development.
2. Get a working knowledge of using extreme automation through XP programming practices of test first development, refactoring and automating test case writing.
3. It is important to comprehend the need to automate the software development lifecycle stages through DevOps. Gain an understanding of the capabilities required to implement DevOps, continuous integration and continuous delivery practices.

4. Configure the web application and Version control using Git using Git commands and version control operations.
5. Configure a static code analyzer which will perform static analysis of the web application code and identify the coding practices that are not appropriate. Configure the profiles and dashboard of the static code analysis tool.
6. Write a build script to build the application using a build automation tool like Maven. Create a folder structure that will run the build script and invoke the various software development build stages. This script should invoke the static analysis tool and unit test cases and deploy the application to a web application server like Tomcat.
7. Configure the Jenkins tool with the required paths, path variables, users and pipeline views.
8. Configure the Jenkins pipeline to call the build script jobs and configure to run it whenever there is a change made to an application in the version control system. Make a change to the background color of the landing page of the web application and check if the configured pipeline runs.
9. Create a pipeline view of the Jenkins pipeline used in Exercise 8. Configure it with user defined messages.
10. In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for static analysis of code.
11. In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for static unit testing.
12. In the configured Jenkins pipeline created in Exercise 8 and 9, implement quality gates for code coverage.

Additional Practice:

1. To fetch and synchronize git repository
2. To perform basic branching and merging in Git
3. To implement form validation

Reference Books:

1 Learning Continuous Integration with Jenkins: A beginner's guide to implementing Continuous Integration and Continuous Delivery using Jenkins - Nikhil Pathania ,Packt publication [https://www.amazon.in/Learning-Continuous-Integration-Jenkins-Pat, ISBN: 978-1785284830

2 Jenkins 2 – Up and Running: Evolve Your Deployment Pipeline for Next Generation Automation – Brent Laster, O'Reilly publication, ISBN: 978-1491979594 [https://www.amazon.in/Jenkins-2-Running-Brent-Laster/dp/1491979593]

Web Links:

1 https://www.geeksforgeeks.org/what-is-ci-cd/

2 https://github.com/nkatre/Free-DevOps-Books-1/blob