Skill Enhancement Courses (SEC)

Course Code	Course Name	Level	L	T	P	C	CIE	SEE	Total	Pre-requisite
241CS016	Introduction to MERN Stack Development	IC			2	2	50	50	100	-
241CS017	Object Oriented Analysis & Design using UML	IC			2	2	50	50	100	ASE
241CS018	Advanced MERN Stack Development	AC			2	2	50	50	100	IMSD
241IT008	Bigdata Spark	AC			2	2	50	50	100	AMSD
241CS019	CI/CD using DevOps	AC			1	1	100	-	100	-
Total					**9**	**9**				

Introduction to MERN Stack Development
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 241CS016	**0**	**0**	**2**	**2**

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

- **CO1:** Make use of HTML elements and their attributes for designing static web pages.
- **CO2:** Build a web page by applying appropriate CSS styles to HTML elements.
- **CO3:** Experiment with JavaScript to develop dynamic web pages and validate forms.
- **CO4:** Build a basic web server using Node.js and also working with Node Package Manager (NPM)
- **CO5:** Make use of Typescript to optimize JavaScript code using strict type checking

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 and Images

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, in such a way that, rather than placing large images on a page, the preferred technique is to use thumbnails by setting the height and width parameters to something like to 100*100 pixels. Each thumbnail image is also a link to a full sized version of the image. Create an image gallery using this technique.

2. HTML Tables, Forms and Frames

a. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan)
b. Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use <caption> tag to set the caption to the table & also use cell spacing, cell padding, border, rowspan, colspan etc.).
c. 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, checkboxes, radio buttons, list boxes using <select> & <option> tags, <text area> and two buttons ie: submit and reset. Use tables to provide a better view).
d. Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. (Note: first frame image, second frame paragraph, third frame hyperlink. And also make sure of using "no frame" attribute such that frames to be fixed).

3. HTML 5 and Cascading Style Sheets, Types of CSS

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 apply different types (or levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).

4. Selector forms

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. Pseudo-class selector
iv. Pseudo-element selector
v. Attribute selector

5. CSS with Color, Background, Font, Text and CSS Box Model

a. Write a program to demonstrate the various ways you can reference a color in CSS.
b. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.
c. Write a program using the following terms related to CSS font and text:
i. font-size ii. font-weight iii. font-style iv. text-decoration
v. text-transformation vi. text-alignment
d. Write a program, to explain the importance of CSS Box model using
i. Content ii. Border iii. Margin iv. padding

6. Applying JavaScript - internal and external, I/O, Type Conversion

a. Write a program to embed internal and external JavaScript in a web page.
b. Write a program to explain the different ways for displaying output.
c. Write a program to explain the different ways for taking input.
d. Create a webpage which uses prompt dialogue box to ask a voter for his name and age. Display the information in table format along with either the voter can vote or not.

7. Javascript Pre-defined and User-defined Objects

a. Write a program using document object properties and methods.
b. Write a program using window object properties and methods.
c. Write a program using array object properties and methods.
d. Write a program using math object properties and methods.
e. Write a program using string object properties and methods.
f. Write a program using regex object properties and methods.
g. Write a program using date object properties and methods.
h. Write a program to explain user-defined object by using properties, methods, accessors, constructors and display.

8. Javascript Conditional Statements and Loops

a. 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".
b. Write a program to display **week** days using switch case.
c. Write a program to print 1 to 10 numbers using for, while and do-while loops.
d. Write a program to print data in object using for-in, for-each and for-of loops

9. Javascript Functions and Events

a. Design a appropriate function should be called to display
   i. Factorial of that number
   ii. Fibonacci series up to that number
b. Design a HTML having a text box and four buttons named Factorial, Fibonacci, When a button is pressed an appropriate function should be called to display
   i. Factorial of that number
   ii. Fibonacci series up to that number
c. Write a program to validate the following fields in a registration page
   i. Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters)
   ii. Mobile (only numbers and length 10 digits)
   iii. E-mail (should contain format like xxxxxxx@xxxxxx.xxx)

10. Node.js

a. Write a program to show the workflow of JavaScript code executable by creating web server in Node.js.
b. Write a program to transfer data over http protocol using http module.
c. Create a text file src.txt and add the following content to it. (HTML, CSS, Javascript, Typescript, MongoDB, Express.js, React.js, Node.js)
d. Write a program to parse an URL using URL module.
e. Write a program to create an user-defined module and show the workflow of Modularization of application using Node.js

11. Typescript

a. Write a program to understand simple and special types.
b. Write a program to understand function parameter and return types.
c. Write a program to show the importance with Arrow function. Use optional, default and REST parameters.
d. Write a program to understand the working of typescript with class, constructor, properties, methods and access specifiers.

Additional Practice:

1. Write a CSS program, to apply 2D and 3D transformations in a web page
2. Design a Web page with new features of HTML file and CSS3.
3. Design a to-do list application using Javascript.

Note: The student must Complete & Submit a HTML5, JavaScript, TypeScript & NodeJs Certificate Courses offered by Infosys Spring board at the end of the Practice Session.

Text Books:

1 Programming the World Wide Web, Robet W Sebesta, Pearson, 7th Edition. ISBN 13 9789332518827
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/en/app/toc/lex_32407835671946760000_shared/overview (Node.js)
4 https://infyspringboard.onwingspan.com/en/app/toc/lex_9436233116512678000_shared/overview (Typescript)

Object Oriented Analysis & Design using UML

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

	L	T	P	C
Course Code: 241CS017	**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	2	3	1							
CO2	2	3	2								
CO3	3	2	2	1	3						
CO4	1	2									
CO5	2	1	1	3	1						

Mapping of Course Outcomes with Program Specific Outcomes:

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

Practice:

List of Case Studies:

ATMApplication.
Library ManagementSystem.
Online Book Shop.
Customer SupportSystem.
Point ofSale.
Familiarization with RationalRose

1. For each case study:
   1.1) Identify and analyze events
   1.2) Identify Usecases
2. For each case study:
   2.1) Develop event table
   2.2) Identify &analyze domain classes
3. For each case study:

3.1) Represent use cases and a domain class diagram using Rational Rose
3.2) Develop CRUD matrix to represent relationships between use cases and problem domain classes

4. For each case study:
   4.1) Develop Use case diagrams
   4.2) Develop elaborate Use case descriptions &scenarios
5. For each case study
   5.1) develop system sequence diagrams
6. For each case study:
   6.1) Develop highlevel sequence diagrams for each usecase
   6.2) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing interactions among all the threelayerobjects
7. For each case study:
   7.1) Develop highlevel sequence diagrams for each usecase1
   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 usecase
   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 Collaboration diagrams for each usecase
   9.2) Develop detailed collaboration diagrams for each use case showing interactions among all the threelayerobjects
10. For each case study:
    10.1) Develop Use case Packages.
    10.2) Develop component diagrams.
    10.3) Develop sample diagrams for other UML diagrams state chart diagrams, activity diagrams and deployment diagrams.
11. 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,AddisonWesley. 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/

Advanced MERN Stack Development
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 241CS018	**0**	**0**	**2**	**2**

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

- **CO1:** Make use of router, template engine and authentication using sessions to develop application in ExpressJS.
- **CO2:** Build a single page application using RESTful APIs in ExpressJS.
- **CO3:** Make use of components, props, stats and render data in ReactJS.
- **CO4:** Apply router and hooks in designing ReactJS application.
- **CO5:** Make use of MongoDB queries to perform CRUD operations on document database.

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		2			1	2		2
CO4	2	3	2		2			1	2		2
CO5	2	2	3		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. **ExpressJS – Routing, HTTP Methods, Middleware.**
   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 to show the working of middleware.

- **ExpressJS- Routing, HTTP Methods, MiddleWare.**
  https://www.codechef.com/learn/course/nodejs/VWHEAQ/problems/NEZHKD07
  https://www.codechef.com/learn/course/nodejs/VWHEAQ/problems/NEZHKD03
  https://www.codechef.com/learn/course/nodejs/MMVQCY/problems/XQACPR01
  https://www.codechef.com/learn/course/nodejs/LNZVHH/problems/TJEKQL07
  https://www.codechef.com/learn/course/nodejs/LNZVHH/problems/TJEKQL13
  https://www.codechef.com/learn/course/nodejs/LNZVHH/problems/TJEKQL16

2. **ExpressJS – Templating, Form Data**
   a. Write a program using templating engine.
   b. Write a program to work with form data.

- **ExpressJS – Templating, Form Data**
  https://www.codechef.com/learn/course/nodejs/VWHEAQ/problems/NEZHKD04
  https://www.codechef.com/learn/course/nodejs/VWHEAQ/problems/NEZHKD09
  https://www.codechef.com/learn/course/nodejs/VWHEAQ/problems/NEZHKD12
  https://www.codechef.com/learn/course/nodejs/VWHEAQ/problems/NEZHKD07
  https://www.codechef.com/learn/course/nodejs/VWHEAQ/problems/NEZHKD11
  https://www.codechef.com/learn/course/nodejs/VWHEAQ/problems/NEZHKD12

3. **ExpressJS – Cookies, Sessions, Authentication**
   a. Write a program for session management using cookies and sessions.
   b. Write a program for user authentication.

- **ExpressJS – Cookies, Sessions, Authentication**
  https://www.codechef.com/learn/course/nodejs/OGHSPI/problems/BIVSBQ01
  https://www.codechef.com/learn/course/nodejs/OGHSPI/problems/BIVSBQ02
  https://www.codechef.com/learn/course/nodejs/OGHSPI/problems/BIVSBQ03
  https://www.codechef.com/learn/course/nodejs/OGHSPI/problems/BIVSBQ04
  https://www.codechef.com/learn/course/nodejs/OGHSPI/problems/BIVSBQ05
  https://www.codechef.com/learn/course/nodejs/OGHSPI/problems/BIVSBQ06

4. **ExpressJS – Database, RESTful APIs**
   a. Write a program to connect MongoDB database using Mangoose and perform CRUD operations.
   b. Write a program to develop a single page application using RESTful APIs.

- **ExpressJS – Database, RESTful APIs**
  https://www.codechef.com/learn/course/nodejs/KVHFAE/problems/NIPFVX01
  https://www.codechef.com/learn/course/nodejs/KVHFAE/problems/NIPFVX02
  https://www.codechef.com/learn/course/nodejs/KVHFAE/problems/NIPFVX04
  https://www.codechef.com/learn/course/nodejs/KVHFAE/problems/NIPFVX08
  https://www.codechef.com/learn/course/nodejs/QVFGYX/problems/JBFUYU01
  https://www.codechef.com/learn/course/nodejs/QVFGYX/problems/JBFUYU02
  https://www.codechef.com/learn/course/nodejs/QVFGYX/problems/JBFUYU03

5. **ReactJS – Render HTML, JSX, Components – function & Class**
   a. Write a program to render HTML to a web page.
   b. Write a program for writing markup with JSX.
   c. Write a program for creating and nesting components (function and class).

- **ReactJS – Render HTML, JSX, Components – function & Class**
  https://www.hackerrank.com/challenges/item-listmanager/problem?IsFullScreen=true

6. **ReactJS – Props and States, Styles, Respond to Events**
   a. Write a program to work with props and states.
   b. Write a program to add styles (CSS & Sass Styling) and display data.
   c. Write a program for responding to events.

- **ReactJS – Props and States, Styles, Respond to Events**
  https://www.hackerrank.com/challenges/code-review-feedback/problem?IsFullScreen=true

7. ReactJS – Conditional Rendering, Rendering Lists, React Forms

a. Write a program for conditional rendering.
b. Write a program for rendering lists.
c. Write a program for working with different form fields using react forms.

- **ReactJS – Conditional Rendering, Rendering Lists, React Form**
  https://www.hackerrank.com/challenges/employee-validation/problem?isFullScreen=true

8. ReactJS – React Router, Updating the Screen

a. Write a program for routing to different pages using react router.
b. Write a program for updating the screen.

- **ReactJS – React Router, Updating the Screen**
  https://www.hackerrank.com/challenges/patient-medical-records/problem?isFull Screen=true

9. ReactJS – Hooks, Sharing data between Components

a. Write a program to understand the importance of using hooks.
b. Write a program for sharing data between components.

- **ReactJS – Hooks, Sharing data between Components**
  https://www.hackerrank.com/challenges/cryptorank-exchange/problem?isFullScreen =true

10. ReactJS Applications – To-do list and Quiz

a. Design to-do list application.

11. MongoDB – Installation, Configuration, CRUD operations

a. Install MongoDB and configure ATLAS
b. Write MongoDB queries to perform CRUD operations on document using insert(), find(), update(), remove()

12. MongoDB – Databases, Collections and Records

a. Write MongoDB queries to Create and drop databases and collections.
b. Write MongoDB queries to work with records using find(), limit(), sort(), createIndex(), aggregate().

Cornerstone Project:
Project:1
1.Disaster Management & Emergency Alert System
A real-time emergency response system for reporting incidents and notifying authorities.
Modules

- Real-time location-based incident reporting
- SOS alert system
- Authority dashboard to verify/respond to cases
- Push notifications & alert zones
- Incident history & analytics

Project:2
Smart Classroom Scheduler with Conflict Detection
An intelligent platform to automate classroom/lab scheduling and avoid timetable clashes.
Modules

- Classroom & timetable creation
- Faculty/course allocation
- Automatic conflict detection using algorithms
- Suggestions for available rooms & slots
- Calendar view, notifications & admin dashboard

Project:3
Student Course Registration and Report Generator
Modules

- User Authentication
  - Student login
  - Admin login
- Student Profile Management
  - View/update personal details
  - Academic details
- Course Catalog
  - View available courses
  - Search/filter courses
  - Course details page
- Course Registration
  - Add/drop courses
  - Registration limit validation
  - Conflict detection (optional)
- Report Generation
  - Registered courses report (PDF/Excel)
  - Semester-wise summary
  - Downloadable registration slip
- Admin Panel
  - Course management
  - Student registration monitoring
  - Report analytics

Project:4
Feedback Collection and Analysis System
Modules

- Login & Role Management
  - Student login
  - Faculty/admin login
- Feedback Form Management
  - Create/edit feedback forms
  - Add questions (text, rating, MCQ)
  - Assign forms to courses
- Feedback Submission
  - Student dashboard

    - Multiple feedback forms
    - Anonymous submission
- Feedback Analytics
    - Rating calculations
    - Charts & visualizations
    - Sentiment analysis (optional AI module)
- Admin Dashboard
    - View feedback summaries
    - Export to PDF/Excel
    - Compare feedback across semesters

Project:5
Online Portfolio Builder
Modules

- User Authentication
    - Register/login
    - Forgot password
- Portfolio Template Selection
    - Template gallery
    - Live preview
- Portfolio Editor
    - Add/update personal info
    - Skills section
    - Projects section
    - Certificates & achievements
    - Social links
- File & Media Upload
    - Project images
    - Resume upload
- Portfolio Publishing
    - Auto-generate public portfolio URL
    - Shareable link
- Admin Panel (Optional)
    - Manage templates
    - User management

Project:6
Real-Time Chat Application
Modules

- User Authentication
    - Register/login
    - User presence (online/offline)
- Real-Time Messaging (Socket.IO)
    - One-to-one chat
    - Typing indicator
    - Message status (sent, delivered, seen)
- Group Chat
    - Create groups

- Add/remove participants
- Media Sharing
  - Images, files, emojis
  - Preview before sending
- Chat History
  - Message storage in DB
  - Search chat messages
- Notifications
  - New message alerts
  - Browser notifications

Project:7

E-commerce Product Listing and Cart System

Modules

- User Authentication
  - Customer login/register
  - Admin login
- Product Catalog
  - Product listing with categories
  - Product details page
  - Search & filters
- Shopping Cart
  - Add/remove items
  - Update quantity
  - Auto-price calculation
- Checkout (Basic)
  - Address form
  - Order summary
  - Payment simulation
- Order Management
  - User order history
  - Admin order tracking
- Admin Panel
  - Product CRUD (add/edit/delete)
  - Category management
  - Stock management

Project:8

College Event Management System

A web portal where students can view, register, and manage college events.

Modules

- User Authentication & Roles Module
  - Student login
  - Organizer login
  - Admin login
  - Password reset
  - Role-based access control
- Dashboard Module
  - Student dashboard (upcoming events, registered events)

    - Organizer dashboard (event stats, registrations)
    - Admin dashboard (overall analytics, approvals)
- Event Creation & Management Module
    - Create new event
    - Upload event details (date, venue, poster, rules)
    - Set registration deadline
    - Edit / update event details
    - Event status (upcoming / ongoing / completed)
- Event Listing & Browsing Module
    - Display all events
    - Search events (by name, category)
    - Filter events (technical, cultural, sports)
    - Event details page
- Event Registration Module
    - Register for an event
    - Register as a team or individual
    - Auto-generate participant ID
    - Registration confirmation email/SMS (optional)
    - Manage registered participants
- Participant Management Module
    - View participant list
    - Download participant list (Excel/PDF)
    - Update participant status
- Payment Management Module (Optional for students)
    - Online payment gateway integration
    - Payment history
    - Refund handling
    - Generate receipts
- Notification & Communication Module
    - Email/SMS alerts for registrations
    - Notifications for event updates
    - Admin announcements
- Event Results & Certificate Module
    - Upload winners list
    - Generate digital certificates
    - Allow participants to download certificates
    - Score/points entry (if competition-based)
- Feedback & Rating Module
    - Student feedback after event
    - Event rating
    - Organizer response
    - Feedback analytics
- Admin Management Module
    - User management (add/delete organizers)
    - Event approval workflow
    - System settings
    - Report generation

- Reports & Analytics Module
  - Event-wise reports
  - Registration statistics
  - Participant demographics
  - Downloadable reports (PDF/Excel)

Project:9
Hostel Management Portal
System for managing hostel room allotment, complaints, and notices.
Modules:
- Room allocation
- complaint submission
- admin approval
- status tracking.

Project:10
Online Examination Platform
Web-based system for conducting multiple-choice exams for students.
Modules:
- Question management
- timer-based quiz
- result generation
- analytics.

Project:11
Student Placement Tracker
Platform for training & placement updates — students can view drives and apply.
Modules:
- Job posting (admin)
- student registration
- eligibility filtering
- status update.

Project:12
Faculty Leave Management System
Portal for faculty to apply and track leaves, with admin (HOD) approval.
Modules:
- Login
- apply leave
- leave history
- approval workflow
- email notifications.

Project:13
Digital Library System
Web app for viewing, searching, and issuing e-books and PDFs.
Modules:
- Search/filter
- user authentication
- admin upload
- download counter.

Project:14
Food Delivery Web App
Simplified food ordering system for campus canteen.
Modules:

- Menu display
- add to cart
- checkout
- order history
- admin order management.

Project:15
Attendance Monitoring System
System for faculty to mark attendance and students to view their records.
Modules:

- Class creation
- attendance marking
- reports
- analytics dashboard.

Project:16
Complaint Management Portal
Platform for students to raise issues related to academics or facilities.
Modules:

- Complaint submission
- Tracking
- admin resolution
- feedback.

Reference Books:

1 Programming the World Wide Web, 7th Edition, Robet W Sebesta, Pearson. ISBN 13 9789332518827

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/web/en/app/toc/lex_auth_013843290195951616350 73_shared/o (Express Js)
2 https://infyspringboard.onwingspan.com/web/en/app/toc/lex_10648877150323546000_shared/overview(
3 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 cornerstone 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 the cornerstone project development.
- The 50 marks will be awarded based on project implementation, oral presentation, 10-minute video presentation, report and viva voce.
- Each student in a team must demonstrate their work for a minimum of 2 minutes in the video presentation.
- Finally, the Source code of the cornerstone Project has to be pushed into the Students GitHub repository.

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

	L	T	P	C
Course Code: 241IT008	**0**	**0**	**2**	**2**

Course Outcomes:

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

CO1: Demonstrate Hadoop Installation process.
CO2: Illustrate Hadoop architecture
CO3: Compare Bigdata models
CO4: Implement various Scalable Algorithms
CO5: Illustrate Applications of Bigdata

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	3	2	2	1	2	1					
CO2	2	2	2	1	2						
CO3	2	2	3	1	2	1					
CO4	2		2	1	3						
CO5	2		2	1	3	1					

Mapping of Course Outcomes with Program Specific Outcomes:

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

Practice:

1. Set up and Configuration Hadoop Using Cloudera Creating a HDFS System with minimum 1 Name Node and 1 Data Nodes HDFS Commands Self-Learning Topics: Set up Hadoop in Linux Environment.
2. Map Reduce Programming Examples Word Count. Union, Intersection and Difference. Matrix Multiplication. Self-Learning Topics: Natural Join Programming Example.
3. Implement an iterative PageRank graph algorithm in MapReduce.
4. Perform an efficient semi-join in MapReduce. Hint: Perform a semi-join by having the mappers load a Bloom filter from the Distributed Cache, and then filter results from the actual MapReduce data source by performing membership queries against the Bloom filter to determine which data source records should be emitted to the reducers.
5. Hive: Introduction Creation of Database and Table, Hive Partition, Hive Built in Function and Operators, Hive View and Index. Self-Learning Topics: Configure Hive Metastore to MySQL

6. Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes
7. Pig: Pig Latin Basic Pig Shell, Pig Data Types, Creating a Pig Data Model, Reading and Storing Data, Pig Operations Self-Learning Topics:
8. Analytics at Large Scale: Libraries of algorithms include SparkMLlib, H20; integrations with TensorFlow and PyTorch;
9. Visualization: Connect to data, Build Charts and Analyze Data, Create Dashboard, Create Stories using Tableau Self-Learning Topics: Tableau using web.
10. Mongo DB: Installation and Creation of database and Collection CRUD Document: Insert, Query, Update and Delete Document. Self-Learning Topics: HBASE Commands

Additional Practice:

1. Spark: RDD, Actions and Transformation on RDD , Ways to Create -file, data in memory, other RDD. Lazy Execution, Persisti RDD Self-Learning Topics: Machine Learning Algorithms like K-Means using Spark
2. Spark joins: Consider a scenario where 2 datasets of a leading retail client to be joined with one another using Spark joins. Customer dataset: Sales dataset: Schema Details: 101 ravi 1 102 keerth 2 101 Syam 1 101 Geetha 1 103 Dawn 3 101 ravi 1 102 keerth 2 101 Syam 1 101 Geetha 1 103 Dawn 3) 66 Customer schema (customer id, customer name, product id) Sales Schema (product id, product name and price) Join both datasets with common key Product id and print customer id, customer name, product name and price.
3. Matrix-Vector Multiplication by Map Reduce
4. Implement an iterative PageRank graph algorithm in MapReduce

Note: The student must Complete & Submit a Big Data 101 & Big Data - 201 Certificate Courses offered by Infosys Spring board at the end of the Practice Session.

Text Books:

1 Understanding Big data, Chris Eaton, Dirk Deroos et al., McGraw Hill, ISBN : 9789339221270.
2 Professional Hadoop Solutions, Boris Lublin sky, Kevin t. Smith, Alexey Yakubovich, Wiley, ISBN: 9788126551071.

Reference Books:

1 Spark in Action, Marko Bonaci and Petar Zecevic, Manning, ISBN: 978-1617292606
2 Py Spark SQL Recipes: With HiveQL, Data frame and Graph frames, Raju Kumar Mishra and Sundar Rajan Raman, A press Media, ISBN: 978-1484243343

Web Links:

1 https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01256841991858585686_shared/overview
2 https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01258388119638835242_shared/overview
3 https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_012605268423008256169 2_shared/overview

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

	L	T	P	C
Course Code: 241CS019	**0**	**0**	**1**	**1**

Course Outcomes:

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

CO1: Summarize the life cycle of agile software development and Develop the fundamental programs using Devops
CO2: Analyze the adoption of Devops in Projects and Process
CO3: Demonstrate the types of continuous integration and continuous delivery in Devops
CO4: Build an automated CI/CD pipeline using a stack of tools
CO5: 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

Mapping of Course Outcomes with Program Specific Outcomes:

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

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