Re-Accredited B++ 2.66 CGPA by NAAC
VEER NARMAD SOUTH GUJARAT UNIVERSITY
University Campus, Udhna-Magdalla Road, SURAT - 395 007, Gujarat, India.
વીર નર્મદ દક્ષિણ ગુજરાત યુનિવર્સિટી
યુનિવર્સિટી કેમ્પસ, ઉધના-મગદલ્લા રોડ, સુરત - ૩૯૫ ૦૦૭, ગુજરાત, ભારત.
Tel : +91 - 261 - 2227141 to 2227146. Toll Free : 1800 2333 011. Digital Helpline No.- 0261 2388888
E-mail : info@vnsgu.ac.in. Website : www.vnsgu.ac.in

–: પરિપત્ર :–

યુનિવર્સિટી સંલગ્ન બી.એસસી. કોમ્પ્યુટર સાયન્સ વિષય ચલાવતી તમામ કોલેજોનાં આચાર્યશ્રીઓને જણાવવાનું કે, શૈક્ષણિક વર્ષ ૨૦૨૬–૨૭ થી અમલમાં આવનાર B.Sc. (Computer Science) (Honours) Sem.- 7 & 8 નો પેટાસમિતિ દ્વારા તૈયાર કરવામાં આવેલ અભ્યાસક્રમ કોમ્પ્યુટર સાયન્સ વિષયની અભ્યાસ સમિતિના ચેરમેનશ્રીએ અભ્યાસ સમિતિ વતી મંજૂર કરી કોમ્પ્યુટર સાયન્સ ફેકલ્ટીને કરેલ ભલામણ કોમ્પ્યુટર સાયન્સ ફેકલ્ટીની તા.૨૯/૦૪/૨૦૨૬ની સભાના ઠરાવ ક્રમાંકઃ૧૭ થી મંજૂર કરી એકેડેમિક કાઉન્સિલને કરેલ ભલામણને એકેડેમિક કાઉન્સિલની તા.૦૭/૦૫/૨૦૨૬ની સભાનાં ઠરાવ ક્રમાંકઃ૫૮ થી મંજૂર કરેલ છે. જેનો અમલ કરવા આથી જાણ કરવામાં આવે છે.

બિડાણઃ ઉપર મુજબ

ક્રમાંકઃઓથો./પરિપત્ર/સિલેબસ/૧૦૦૪૯/૨૦૨૬
તા.૨૦–૦૫–૨૦૨૬

કુલસચિવ

પ્રતિ,
૧) યુનિવર્સિટી સંલગ્ન બી.એસસી. કોમ્પ્યુટર સાયન્સ વિષય ચલાવતી કોલેજોનાં આચાર્યશ્રીઓ.
.....આપશ્રીની કોલેજના સંબંધિત શિક્ષકોને જાણ કરી અમલ કરવા સારૂ.
૨) ડીનશ્રી, કોમ્પ્યુટર સાયન્સ વિદ્યાશાખા.
૩) પરીક્ષા નિયામકશ્રી, પરીક્ષા વિભાગ, વીર નર્મદ દ. ગુ. યુનિવર્સિટી, સુરત.
.....તરફ જાણ તેમજ અમલ સારૂ.

Veer Narmad South Gujarat University, Surat

Computer Science and Information Technology Faculty

Syllabus for(Semester-VII and Semester-VIII)

of

B.Sc.(Computer Science) (Honours)

As per NEP- 2020

To be implemented from Academic Year: June, 2026-2027

Veer Narmad South Gujarat University, Surat
Bachelor of Science Computer Science
(B.Sc.(Computer Science)(Honours))
Under the Faculty of
Computer Science and Information Technology

Name of Program:	Bachelor of Science Computer Science(Honours)
Abbreviation:	B.Sc (Computer Science)(Honours): Four-year Integrated Program with Multi-Level Entry and Exit option
Multi-level Exit Criteria:	i) Under Graduate Certificate in Computer Science: If the student wish to exit after completion of First year (Semester-1 and Semeter-2) without any back- log and secure additional 4 credits from work based skill oriented university approved courses /vocational courses / summer internship / Apprenticeship in addition to 6 credits from skill-based courses earned during first and second semester. ii) Diploma in Computer Science: If the student wish to exit after completion of Second year (Semester-1 to Semeter-4) without any back-log and secure additional 4 credits from work based skill oriented university approved courses/vocational courses/summer internship/Apprenticeship offered at end of first or second year in addition to 6 credits from skill-based courses earned during first four semesters. iii) Bachelor of Science Computer Science B.Sc (Computer Science): If wish to exit after completion of Third year (Semeste-1 to semester-6) without any back-log and secure additional 4 credits from work based skill oriented university approved courses/vocational courses/summer internship/Apprenticeship offered at end of first or second year in addition to 6 credits from skill-based courses earned during first four semesters.
Multi-Level Entry Criteria:	As per the norms of the Veer Narmad South Gujarat University.
Duration:	4 year of Bachelor of Science Computer Science (Honours) degree program with multilevel exit options at 1st, 2nd and 3rd Year to obtain Certificate, Diploma, Degree and Honours Degree in Computer Science respectively.
Eligibility:	Candidate must have passed standard 12th (H.S.C.) Examination in Science (Any from Gujarat Higher Secondary Board (G.H.S.E.B.) or any other equivalent board (C.B.S.E. / I.C.S.E. etc. which must be approved and possess equivalence certificate from Veer Narmad South Gujarat University) with English as one of the subject.
Objective of the Program:	Bachelor of Science Computer Science (Honours) is undergraduate degree program in computer Science area. Objective of the program is to open a channel of admission for courses in the field of Computer Science, Applications and all relevant fields of information technologies to build career for students who have completed standard 12th (H.S.C.)

	and are interested in taking computing/computer Application and Information Technology as a career. Main objective is to equip the students with strong foundation in computer programming languages, coding, database handling, software application developments, problem-solving skills and development of analytical and logical skills. The focus is to introduce various programming languages on different platforms and operating systems, interaction with databases available on various platforms, software testing, development and deployment techniques. It also aims to provide knowledge in latest trends and advancements in field of computer technologies. The program caters to the needs of the students aspiring to excel in the field of computer science, applications and technologies. The program is designed to develop computer professionals versatile in almost all field of computer application. It also aims to enhance communication and interpersonal skills.
Program Outcome:	**PO1:** Ability to analyze a problem, identify and define the Computing requirements appropriate to its solution. **PO2:** Enhancing the problem solving, logical, reasoning and analysis capabilities of a problem and integrate the ability with the coding using specific computer programming languages. **PO3:** To generate Understanding regarding the core and fundamental ideas about the computer platforms, operating systems, software design concepts, networking concepts and advanced and emerging technologies. **PO4:** Design, implement and evaluate a computer-based system, processing, component or program to meet desired goal with the help of various programming languages, application software, packages, tools, databases on various platforms. **PO5:** An ability to apply design and development principles in construction of software systems of varying complexity using various algorithmic principles, modeling, coding and design of computer-based systems. **PO6:** Prepare the aspiring students to become computer software professionals who can work in corporate/software industry at entry to advanced level as well as independent developers. Overall, the program outcomes aim to produce graduates who are: (a) competent in computer application, development and design. (b) Adapt to changing technology and industry trends. (c) Can make significant contributions to the software applications coding, designing, database managements, testing, deployments and ready to adapt any upcoming technologies.
Program Specific Outcome:	**PSO1:** Developing understanding about the fundamentals of core concepts of logic developments, critical thinking and problem solving capabilities. Emphasis on effective communication. **PSO2:** Improving analytical and applied concepts using various technologies, coding concepts and implementation of coding to solve the problems. **PSO3:** Development of team building concepts and working in team with

	positive approach, enhancing the mindset to contribute as an individual to the team. Improving interpersonal skills. **PSO4:** Improving student's Understanding related to technical problems and enhancing their capabilities to address the problems to turn into solutions through various possible ways by enhancing critical thinking ability. **PSO5:** Develop students to capabilities for self-learning, skill development through self-practicing and problem solving abilities. **PSO6:** Develop students to address and work on the real-world problems as an individual and as part of team. Understand the business problems and ability to work on their solutions by applying various software technologies. **PSO7:** To enhance development skills at various level including problem analysis, data analysis, logical and critical analysis of the problems and implementing the solutions by imparting various recent and upcoming technologies. **PSO8:** Enhance the passion among the students for updating knowledge, innovative ideas, up skilling and implementing the knowledge in applied areas and research areas by understanding the real world problems, addressing the real world problems and their possible solutions that lead to build a successful Professional career.
PO and PSO mapping:	
Medium of Instruction:	English
Program Structure:	Semester-wise Breakup of the course is given as follows:

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8
PO1								
PO2								
PO3								
PO4								
PO5								
PO6								

Veer Narmad South Gujarat University, Surat

Program Structure: Fourth Year B.Sc. (Computer Science) Honors

SEMESTER-VII

(w.e.f. Academic Year June, 2026)

Course Code	Course Title	Course Category	Level of Course	Course Credits	Teaching Hours/Week	
				Th. + Pract.	Theory	Practical/ Field Work/ Project/ Internship
CS-701	DevOps & Production Deployment	Minor Course	300-499	4	4	0
CS-702-01 OR CS-702-02	Big Data Technologies & Processing **OR** Advanced Frontend Development	Major Course	400-499	4	2	4
CS-703-01 OR CS-703-02	Machine Learning Engineering & MLOps **OR** Backend Development & Microservices	Major Course	400-499	4	2	4
CS-704-01 OR CS-704-01	Natural Language Processing & Computer Vision **OR** Full-Stack Application Development	Major Course	400-499	4	2	4
CS-705	Project	Major specific Course	400-499	6		12
	Practical (Based on Course Code:702-01/702-02, 703-01,703-02,704-01 & 704-02 Equally divided)	No separate credits allocated for practical. Students will prepare separate practical journals for all 3-courses. The Practical exam/viva-voce will be based on Course 702-01/702-02, 703-01,703-02,704-01 & 704-02				
Other Activities	The student is expected to participate in activities related to National Service Scheme (NCC), National Cadet Corps (NCC), adult education/ literacy initiatives, mentoring school students, Elderly literacy program/ Environment preservation activities and other similar activities.					
Total				**22**	**10**	**24**

Course Code	Course Title	Course Credit	University Exam Type	Exam Duration	External (SEE) Marks	Internal (CCE) Marks	Total Marks
CS-701	DevOps & Production Deployment	4	Theory/ Written	2 Hour	50	50	100
CS-702-01 **OR** CS-702-02	Big Data Technologies & Processing **OR** Advanced Frontend Development	4	Theory/ Written	1 Hour	25	25	100
			Practical	2 Hours	25	25	
CS-703-01 **OR** CS-703-02	Machine Learning Engineering &MLOps **OR** Backend Development & Microservices	4	Theory/ Written	1 Hour	25	25	100
			Practical	2 Hours	25	25	
CS-704-01 **OR** CS-704-02	Natural Language Processing & Computer Vision **OR** Full-Stack Application Development	4	Theory/ Written	1 Hour	25	25	100
			Practical	2 Hours	25	25	
CS-705	Project	6			50	50	100
Total		**22**			**250**	**250**	**500**

For Practical and Project:

- Batch Size – 40 Maximum (Desirable). Maximum 45 students can be accommodated in a batch. Separate batch should be considered if the student strength exceeds 45 numbers.

- Practical includes Practical sessions for course-702-01/702-02,703-01/703-02,704-01 & 704-02. **Minimum** Twelve Practical hours (4 hours for course-702-01/702-02,703-01/703-02,704-01 & 704-02 4 hours each) per week should be allocated per batch.

- The journal must be certified by the concerned faculty and by the Head of the Department, failing which the student will not be allowed to appear for External Practical Examination. Student will submit softcopy/hardcopy of Project duly certified by the internal guide.

Internal/External Evaluation:
CCE(Continuous and Comprehensive Evaluation):To be conducted by college. SEE (Semester End Evaluation) : To be conducted by University.

Major Course: Major discipline is the main focus (Core) dominant subject and the degree will be awarded in that discipline. Students must secure a prescribed number of credits (50% of total credits) through core courses in the major discipline. Students can choose the courses from the pool of courses.

Minor Course: Minor discipline is the broader understanding course beyond the major discipline course. It contains generic- electives for students to choose from the pool of courses. It helps students to gain broader knowledge in addition to relevant major disciplines courses as per their choices. Minor subjects may be from same or different disciplines. Student may make choices according to their

interest/need, from ODL courses also.

Marks: The scale on which the students will be evaluated. The evaluation methodology will be continuous evaluation and the score obtained will reflect in mark-sheet but not considered for SGPA or CGPA.

Major Practical based Subjects: Course 702-01/702-02,703-01/703-02,704-01 & 704-02 are major courses. These courses are carrying 4 credits (2 hours of theory and 4 hours of practical per week). These subjects carry 100 marks of exam weightage (50 theory and 50 practical). External and Internal distribution of marks are in ratio of 50:50 respectively. Students are required to acquire minimum passing marks from theory and practical collectively. Practical exams for course-702-01/702-02 (2 hours duration), 703-01/703-02 (2 hours duration) and course-704-01/704-02 (2 hours duration) will be conducted on same day.

Division of Theory internal marks(CCE):

For courses having 50 marks as Internals:

Class Assignment/Active Learning: 07marks + Home Assignment/ Field Assignment:08 marks + Attendance:10 marks+ Class Test*:25 marks

For courses having 25 marks as Internals :

Class Assignment/ Active Learning:03 marks + Home Assignment/Field Assignment:03 marks + Attendance:04 marks + Class Test*:15 marks

For Practical internal marks(CCE):

Attendance:5 marks + Viva-voce/ Quiz:10 marks + Lab-work Assessment/Practical:10 marks.

Division of Practical External exam marks(SEE):

For 25 marks Externals:

Division of marks are: Exam evaluation: 20 marks + Viva-voce:5 Marks.
Practical examination will be conducted for course code-702-01/702-02,703-01/703-02.704-01/704-02 separately on same day. Students are required to pass in combined head (Theory + Practical) for each course. Students are required to remain present in internal and external theory and practical exams for all course code–701 to 705 mandatorily.

Program Passing Rules:	As per University rules.
Program Fees: (Per Semester) (One time fees and exam fees are additional as prescribed by the university)	Semester Tuition Fees : As per norms of University Semester Laboratory Utilization fees : As per norms of University [Other one time/affiliation/exam fees and other fees under various heads, will be as per the norms of the University.] [The fees for all certificate courses, Skill Enhancement Courses and Value Addition Courses; fees will be as per the prescribed limit for per credit as per the SOP of certificate courses decided by the University.]

Fourth Year B.Sc. (Computer Science) Honors SEMESTER-VIII						
Course Code	**Course Title**	**Course Category**	**Level of Course**	**Course Credits**	**Teaching Hours/Week**	
				Th. + Pract.	**Theory**	**Practical/Field Work/Project/ Internship**
CS-801	Artificial Intelligence & Intelligent Systems	Minor Course	300-499	4	4	0
CS-802	Cloud Native Application Development	Major Course	400-499	4	2	4
CS-803	Emerging Technologies & Industry Practices	Major Course	400-499	4	2	4
CS-804-01 OR CS-804-02	Data Science & Advanced Analytics **OR** Advanced Mobile Application Development	Major Course	400-499	4	2	4
CS-805	On-the-Job Training & Industry Immersion (OJT)	Major specific Course	400-499	6	-	12
	Practical (Based on Course Code:802-01/802-02, 803 & 804 Equally divided)	No separate credits allocated for practical. Students will prepare separate practical journals for all 3-courses. The Practical exam/viva-voce will be based on Course 802,803,804-01/804-02.				
Other Activities	The student is expected to participate in activities related to National Service Scheme (NCC), National Cadet Corps (NCC), adult education/literacy initiatives, mentoring school students, Elderly literacy program/ Environment preservation activities and other similar activities.					
Total				**22**	**10**	**24**

Course Code	Course Title	Course Credit	University Exam Type	Exam Duration	External Marks	Internal Marks	Total Marks
CS-801	Artificial Intelligence & Intelligent Systems	4	Theory/Written	2 Hours	50	50	100
CS-802	Cloud Native Application Development	4	Theory/Written	1 Hour	25	25	100
			Practical	2 Hours	25	25	
CS-803	Emerging Technologies & Industry Practices	4	Theory/Written	1 Hour	25	25	100
			Practical	2 Hours	25	25	
CS-804-01 OR CS-804-02	Data Science & Advanced Analytics **OR** Advanced Mobile Application Development	4	Theory/Written	1 Hour	25	25	100
			Practical	2 Hours	25	25	
CS-805	OJT On-the-Job Training & Industry Immersion	6	Documentation Reporting, Presentation and Viva-voc	Presentation and viva-voce	50	50	100
Total		**22**			**250**	**250**	**500**

For Practical and Project:

- Batch Size – 40 Maximum (Desirable). Maximum 45 students can be accommodated in a batch. Separate batch should be considered if the student strength exceeds 45 numbers.
- Practical includes Practical sessions for course-802,803,804-01 & 804-02
- **Minimum** Twelve Practical hours (4 hours for course-802,803,804-01 & 804-02 , 4 hours each) per week should be allocated per batch.
- The journal must be certified by the concerned faculty and by the Head of the Department, failing which the student will not be allowed to appear for External Practical Examination. Student will submit softcopy/hardcopy of Project duly certified by the internal guide.

Internal/External Evaluation:
CCE(Continuous and Comprehensive Evaluation):To be conducted by college. SEE (Semester End Evaluation) : To be conducted by University.

Major Course: Major discipline is the main focus (Core) dominant subject and the degree will be awarded in that discipline. Students must secure a prescribed number of credits (50% of total credits) through core courses in the major discipline. Students can choose the courses from the pool of courses.

Minor Course: Minor discipline is the broader understanding course beyond the major discipline course. It contains generic- electives for students to choose from the pool of courses. It helps students

to gain broader knowledge in addition to relevant major disciplines courses as per their choices. Minor subjects may be from same or different disciplines. Student may make choices according to their interest/need, from ODL courses also.

Marks: The scale on which the students will be evaluated. The evaluation methodology will be continuous evaluation and the score obtained will reflect in mark-sheet but not considered for SGPA or CGPA.

Major Practical based Subjects: 802,803,804-01 & 804-02 are major courses. These courses are carrying 4 credits (2 hours of theory and 4 hours of practical per week). These subjects carry 100 marks of exam weightage (50 theory and 50 practical). External and Internal distribution of marks are in ratio of 50:50 respectively. Students are required to acquire minimum passing marks from theory and practical collectively. Practical exams for course-802 (2 hours duration), 803 (2 hours duration) and course-804-01 & 804-02 (2 hours duration) will be conducted on same day.

Division of Theory internal marks(CCE):

For courses having 50 marks as Internals:

Class Assignment/Active Learning: 07marks + Home Assignment/ Field Assignment:08 marks + Attendance:10 marks+ Class Test*:25 marks

For courses having 25 marks as Internals :

Class Assignment/ Active Learning:03 marks + Home Assignment/Field Assignment:03 marks + Attendance:04 marks + Class Test*:15 marks

For Practical internal marks(CCE):

Attendance:5 marks + Viva-voce/ Quiz:10 marks + Lab-work Assessment/Practical:10 marks.

Division of Practical External exam marks(SEE):

For 25 marks Externals:

Division of marks are: Exam evaluation: 20 marks + Viva-voce:5 Marks.
Practical examination will be conducted for course code-802,803,804-01/804-02,805. separately on same day. Students are required to pass in combined head (Theory + Practical) for each course. Students are required to remain present in internal and external theory and practical exams for all courses mandatorily.

Program Passing Rules:	As per University rules.
Program Fees: (Per Semester) (One time fees and exam fees are additional as prescribed by the university)	Semester Tuition Fees : As per norms of University Semester Laboratory Utilization fees : As per norms of University [Other one time/affiliation/exam fees and other fees under various heads, will be as per the norms of the University.] [The fees for all certificate courses, Skill Enhancement Courses and Value Addition Courses; fees will be as per the prescribed limit for per credit as per the SOP of certificate courses decided by the University.]

<table>
<tr><th colspan="2">701 - DevOps & Production Deployment</th></tr>
<tr><td>Program Name</td><td>B.Sc. Computer Science (Honours)</td></tr>
<tr><td>Semester</td><td>7</td></tr>
<tr><td>NCrf Credit Level</td><td>6</td></tr>
<tr><td>Course Type</td><td>Minor</td></tr>
<tr><td>Course Subtype</td><td>Skill Development</td></tr>
<tr><td>Subject Type</td><td>Discipline Specific</td></tr>
<tr><td>Course Code</td><td>701</td></tr>
<tr><td>Course Level</td><td>300-499</td></tr>
<tr><td>Course Title</td><td>DevOps & Production Deployment</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Effective From</td><td>Academic Year: 2026-27</td></tr>
<tr><td>Pre- requisite</td><td>Knowledge of basic cloud computing concepts and Computer Network</td></tr>
<tr><td>Course outcome</td><td>CO1: Apply DevOps principles to automate software development workflows
CO2: Analyze version control strategies for team-based development
CO3: Design CI/CD pipelines for application deployment
CO4: Develop containerized applications using Docker
CO5: Implement basic orchestration using Kubernetes concepts
CO6: Evaluate deployment strategies for scalability and reliability
CO7: Build and deploy real-world applications using DevOps tools
CO8: Demonstrate readiness for industry-level DevOps practices..</td></tr>
<tr><td>Course Outcomes (COs) and Mapping with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th></tr>
<tr><td>CO1</td><td>✓</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>CO2</td><td>✓</td><td>✓</td><td>–</td><td>–</td><td>✓</td><td>–</td></tr>
<tr><td>CO3</td><td>✓</td><td>✓</td><td>✓</td><td>–</td><td>✓</td><td>–</td></tr>
<tr><td>CO4</td><td>✓</td><td>–</td><td>✓</td><td>–</td><td>✓</td><td>–</td></tr>
<tr><td>CO5</td><td>✓</td><td>–</td><td>✓</td><td>–</td><td>✓</td><td>–</td></tr>
<tr><td>CO6</td><td>✓</td><td>✓</td><td>✓</td><td>–</td><td>✓</td><td>✓</td></tr>
<tr><td>CO7</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>–</td></tr>
<tr><td>CO8</td><td>–</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
</table>
</td></tr>
<tr><td>Course Content</td><td>Unit–1: DevOps Fundamentals & Culture
1.1 Introduction to DevOps
1.1.1 Concepts of DevOps
1.1.2 DevOps Lifecycle
1.1.3 Benefits of DevOps
1.1.4 Challenges in DevOps</td></tr>
</table>

1.2 DevOps vs Traditional Development Models
1.2.1 Waterfall vs DevOps
1.2.2 Limitations of Traditional Models
1.2.3 Advantages of DevOps Approach
1.3 Agile Methodology & DevOps Integration
1.3.1 Agile Principles
1.3.2 Scrum Basics
1.3.3 Integration of Agile with DevOps
1.4 CI/CD Concepts
1.4.1 Continuous Integration (CI)
1.4.2 Continuous Deployment (CD)
1.4.3 Importance of Automation

Unit–2: Version Control & Collaboration
2.1 Introduction to Version Control Systems
2.1.1 Types of Version Control Systems
2.1.2 Importance in Software Development
2.2 Git Fundamentals
2.2.1 Repository
2.2.2 Commit
2.2.3 Push Operations
2.2.4 Pull Operations
2.3 Branching Strategies
2.3.1 Feature Branch Workflow
2.3.2 Git Flow (Basic Understanding)
2.4 Merge & Conflict Resolution
2.4.1 Merging Branches
2.4.2 Handling Conflicts
2.4.3 Collaboration using GitHub/GitLab
2.5 Best Practices in Code Versioning
2.5.1 Code Management Techniques
2.5.2 Collaboration Workflows
2.5.3 Maintaining Code Quality

Unit–3: Containerization with Docker
3.1 Introduction to Containerization & Virtualization
3.1.1 Concepts of Containerization
3.1.2 Concepts of Virtualization
3.1.3 Differences between Containerization and Virtualization
3.1.4 Advantages of Containerization
3.2 Docker Architecture & Components
3.2.1 Docker Engine
3.2.2 Docker Images
3.2.3 Docker Containers
3.2.4 Docker Registry
3.3 Managing Docker Containers
3.3.1 Creating Containers
3.3.2 Running Containers
3.3.3 Managing Containers
3.4 Dockerfile & Application Deployment
3.4.1 Writing Dockerfile

	3.4.2 Building Docker Images 3.4.3 Deploying Applications using Docker **3.5 Docker Compose Basics** 3.5.1 Multi-container Applications 3.5.2 Configuration using Docker Compose **Unit–4: Kubernetes & Cloud Deployment** **4.1 Introduction to Kubernetes** 4.1.1 Kubernetes Architecture 4.1.2 Need for Orchestration **4.2 Kubernetes Components** 4.2.1 Pods 4.2.2 Nodes 4.2.3 Deployments 4.2.4 Services **4.3 Orchestration & Scaling Concepts** 4.3.1 Load Balancing 4.3.2 Auto Scaling 4.3.3 High Availability **4.4 Cloud Deployment Overview** 4.4.1 Deployment on Cloud Platforms 4.4.2 Overview of AWS 4.4.3 Overview of GCP 4.4.4 Overview of Azure 4.4.5 Benefits of Cloud-based Deployment
Reference Books	1. Docker Deep Dive, Nigel Poulton, Nigel Poulton (Independently Published), 978-1521822807 2. Kubernetes Up & Running (2nd Edition), Brendan Burns; Joe Beda; Kelsey Hightower, O'Reilly Media, 978-1492046530 3. The DevOps Handbook, Gene Kim; Jez Humble; Patrick Debois; John Willis, IT Revolution Press, 978-1942788003 4. Continuous Delivery, Jez Humble; David Farley, Addison-Wesley Professional, 978-0321601919 5. Git Pocket Guide, Richard E. Silverman, O'Reilly Media, 978-1449325862 6. Pro Git (2nd Edition), Scott Chacon; Ben Straub, Apress, 978-1484200773 7. Site Reliability Engineering, Betsy Beyer et al., O'Reilly Media, 978-1491929124 8. Learning Docker, Jeeva S. Chelladhurai; Vinod Singh; Pethuru Raj, Packt Publishing, 978-1786462923 9. DevOps for Beginners, Kunal Kushwaha, Independently Published, 979-8678320148 10. Cloud Native DevOps with Kubernetes, John Arundel; Justin Domingus, O'Reilly Media, 978-1492040767
Teaching Methodology	Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.
Evaluation Method	Internal Assessment: 50 marks Theory = 50 Marks. External Assessment: 50 marks Theory 50 Marks.

	50% Internal assessment - Attendance, class and home assignment, Unit tests. 50% External Assessment. - Written theory exam

CS-702-01 Big Data Technologies & Processing

[Subject code *for Theory*-2611001007011001] ***[Subject code for Practical-2611001007011002]***

Program Name	B.Sc. Computer Science (Honours)					
Semester	7					
NCrf Credit Level	6					
Course Type	Major					
Course Subtype	Skill Development					
Subject Type	Discipline Specific					
Course Code	702-01					
Course Level	400-499					
Course Title	Big Data Technologies & Processing					
Credit	Theory	02 hrs	Practical	04 hrs	Total	04 Cr
Effective From	Academic Year: 2026-27					
Course outcome	CO1 Understand the fundamentals of Big Data ecosystem, distributed computing, and Hadoop architecture. CO2 Apply Apache Spark concepts including RDDs, DataFrames, and PySpark for large-scale data processing. CO3 Analyze and implement NoSQL databases such as MongoDB, Cassandra, and HBase with proper data modeling techniques. CO4 Develop real-time data streaming solutions using Apache Kafka and Spark Streaming. CO5 Design and implement end-to-end big data processing pipelines for real-world applications.					

Course Outcomes (COs) and Mapping with PSOs	**COs**	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**
	CO1	■					
	CO2		■	■			■
	CO3			■	■		■
	CO4	■	■		■	■	
	CO5	■	■			■	■

Course Content

Unit-1: Introduction to Big Data Ecosystem

1.1 Characteristics of Big Data
- 1.1.1 Volume
- 1.1.2 Velocity
- 1.1.3 Variety
- 1.1.4 Veracity

1.2 Distributed Computing Principles
1.3 Hadoop Architecture
1.3.1 HDFS (Hadoop Distributed File System)
1.3.2 MapReduce Paradigm
1.4 YARN Resource Management

Unit-2: Apache Spark for Large-Scale Data Processing

2.1 Spark Architecture
2.1.1 RDDs (Resilient Distributed Datasets)
2.1.2 DataFrames
2.1.3 Datasets
2.2 Transformations vs Actions
2.3 PySpark Implementation
2.4 In-Memory Computing Advantages
2.5 Spark SQL for Structured Data Processing

Unit-3: NoSQL Databases for Big Data

3.1 Comparative Analysis of NoSQL Databases
3.1.1 MongoDB
3.1.2 Cassandra
3.1.3 HBase
3.2 CAP Theorem and Practical Implications
3.3 Data Modeling for NoSQL
3.4 Sharding and Replication Strategies
3.5 Hands-on with MongoDB Aggregation Pipelines

Unit-4: Real-Time Data Streaming & Processing

4.1 Apache Kafka Fundamentals
4.1.1 Producers
4.1.2 Consumers
4.1.3 Topics
4.1.4 Partitions
4.2 Stream Processing Concepts
4.3 Integration of Kafka with Spark Streaming
4.4 Real-Time Analytics Pipeline Development
4.5 Practical Use Cases
4.5.1 IoT Systems
4.5.2 Financial Systems

Reference Books

No.	Title	Author(s)	Publisher	ISBN
1	Big Data: Principles and Best Practices of Scalable Real-Time Data Systems	Nathan Marz, James Warren	Manning	978-1617290343

	2	Hadoop: The Definitive Guide	Tom White	O'Reilly Media	978-1491901632
	3	Learning Spark: Lightning-Fast Big Data Analysis	Holden Karau et al.	O'Reilly Media	978-1449358624
	4	Spark: The Definitive Guide	Bill Chambers, MateiZaharia	O'Reilly Media	978-1491912218
	5	NoSQL Distilled	Pramod J. Sadalage, Martin Fowler	Addison-Wesley	978-0321826626
	6	MongoDB: The Definitive Guide	Kristina Chodorow	O'Reilly Media	978-1449344689
	7	Designing Data-Intensive Applications	Martin Kleppmann	O'Reilly Media	978-1449373320
	8	Kafka: The Definitive Guide	Neha Narkhede, Gwen Shapira, Todd Palino	O'Reilly Media	978-1491936160
	9	Streaming Systems	Tyler Akidau et al.	O'Reilly Media	978-1491983874
	10	Data Science for Big Data Analytics	EMC Education Services	Wiley	978-1118876053
Teaching Methodology	Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.				
Evaluation Method	Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. 50% Internal assessment - Attendance, class and home assignment, Unit tests. - Practical exam, viva-voce, Journal 50% External Assessment. - Written theory exam - Practical exam, Viva-Voce				

CS-702-02 Advanced Frontend Development

Program Name	B.Sc. Computer Science (Honours)					
Semester	7					
NCrf Credit Level	6					
Course Type	Major					
Course Subtype	Skill Development					
Subject Type	Discipline Specific					
Course Code	702-02					
Course Level	400-499					
Course Title	Advanced Frontend Development					
Credit	Theory	02 hr	Practical	04 hr	Total	06 Cr
Effective From	Academic Year: 2026-27					

Course outcome

CO1	Apply modern JavaScript (ES6+) and Typescript concepts to build scalable frontend applications.
CO2	Develop optimized React applications using advanced patterns and performance techniques.
CO3	Implement efficient state management and integrate APIs using Redux Toolkit, REST, and GraphQL
CO4	Design and deploy Progressive Web Applications (PWAs) with offline capabilities and caching strategies.
CO5	Perform frontend testing using modern tools such as Jest, React Testing Library, and Cypress.

Course Outcomes (COs) and Mapping with PSOs

COs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1						
CO2						
CO3						
CO4						
CO5						

Course Content

Unit-1: Modern JavaScript & TypeScript

1.1 ES6+ Features

1.1.1 Destructuring

1.1.2 Spread and Rest Operators

	1.1.3 Arrow Functions 1.1.4 Async/Await **1.2 Type Script Fundamentals** 1.2.1 Basic Types 1.2.2 Interfaces 1.2.3 Generics 1.2.4 Enums **1.3 Type Safety** 1.3.1 Static Typing Concepts 1.3.2 Error Detection and Debugging 1.3.3 Type Checking in Large Applications **1.4 Migration Strategies** 1.4.1 JavaScript to TypeScript Conversion 1.4.2 Gradual Migration Approach 1.4.3 Tooling and Configuration (tsconfig, linters) **Unit-2: Advanced React Patterns & Performance Optimization** **2.1 State Management in React** 2.1.1 React Context API 2.1.2 useReducer Hook 2.1.3 Comparison with useState **2.2 Custom Hooks** 2.2.1 Creating Custom Hooks 2.2.2 Reusability of Logic 2.2.3 Hook Composition **2.3 Performance Optimization** 2.3.1 Code Splitting 2.3.2 Lazy Loading 2.3.3 React.memo 2.3.4 useMemo and useCallback **2.4 Server-Side Rendering (SSR)** 2.4.1 Introduction to SSR 2.4.2 Next.js Basics 2.4.3 Benefits of SSR (SEO, Performance) **Unit-3: State Management & API Integration** **3.1 Redux Advanced Concepts** 3.1.1 Redux Toolkit (RTK) 3.1.2 Slice Creation 3.1.3 Store Configuration **3.2 Middleware** 3.2.1 Redux Thunk 3.2.2 Redux Saga

3.2.3 Middleware Workflow

3.3 API Integration

3.3.1 RTK Query

3.3.2 REST API Integration

3.3.3 Error Handling

3.4 GraphQL

3.4.1 GraphQL Fundamentals

3.4.2 Queries and Mutations

3.4.3 Apollo Client Integration

3.5 REST vs GraphQL

3.5.1 Key Differences

3.5.2 Advantages and Limitations

3.5.3 Use Cases Comparison

Unit-4: Progressive Web Apps & Testing

4.1 Progressive Web Apps (PWA)

4.1.1 Service Workers

4.1.2 Offline Functionality

4.1.3 Caching Strategies

4.2 Web App Manifest

4.2.1 App Installation

4.2.2 Icons and Metadata

4.2.3 Configuration Files

4.3 Push Notifications

4.3.1 Notification APIs

4.3.2 Subscription Management

4.3.3 User Engagement Strategies

4.4 Testing Frameworks

4.4.1 Jest for Unit Testing

4.4.2 React Testing Library

4.4.3 Test Case Design

4.5 End-to-End Testing

4.5.1 Cypress Basics

4.5.2 Writing Test Cases

4.5.3 Automation and Reporting

Reference Books

No.	Title	Author(s)	Publisher	ISBN
1	JavaScript: The Good Parts	Douglas Crockford	O'Reilly Media	978-0596517748
2	Eloquent JavaScript (3rd Edition)	Marijn Haverbeke	No Starch Press	978-1593279509

	3	You Don't Know JS (Series)	Kyle Simpson	O'Reilly Media	978-1491904244
	4	Learning TypeScript	Josh Goldberg	O'Reilly Media	978-1098110338
	5	Programming TypeScript	Boris Cherny	O'Reilly Media	978-1492037651
	6	React Explained	Zac Gordon	Independently Published	978-1797407999
	7	Fullstack React	Anthony Accomazzo et al.	Fullstack.io	978-0991344628
	8	Redux in Action	Marc Garreau, Will Faurot	Manning Publications	978-1617294976
	9	Learning GraphQL	Eve Porcello, Alex Banks	O'Reilly Media	978-1492030713
	10	Testing JavaScript Applications	Lucas Fernandes da Costa	Manning Publications	978-1617297915

Teaching Methodology	Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.
Evaluation Method	Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. 50% Internal assessment - Attendance, class and home assignment, Unit tests. - Practical exam, viva-voce, Journal 50% External Assessment. - Written theory exam - Practical exam, Viva-Voce

CS-703-01: Machine Learning Engineering & MLOps

[Subject code for Theory-2611001007022001] [Subject code for Practical-2611001007022002]

Program Name	B.Sc. Computer Science (Honours)					
Semester	7					
NCrf Credit Level	6					
Course Type	Major					
Course Subtype	Skill Development					
Subject Type	Discipline Specific					
Course Code	703-01					
Course Level	400-499					
Course Title	Machine Learning Engineering &MLOps					
Credit	Theory	02 hr	Practical	04 hr	Total	04 Cr
Effective From	Academic Year: 2026-27					
Course outcome	CO1 Understand and apply advanced machine learning algorithms such as ensemble methods, SVM, and Naive Bayes. CO2 Analyze and implement deep learning models including CNNs, RNNs, LSTMs, and transfer learning techniques. CO3 Evaluate and optimize machine learning models using validation techniques, hyperparameter tuning, and performance metrics. CO4 Apply model interpretability techniques and conduct testing strategies like A/B testing for deployment. CO5 Design and deploy machine learning solutions using MLOps practices, including pipeline automation, containerization, and cloud platforms.					

Course Outcomes (COs) and Mapping with PSOs

COs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1						
CO2						
CO3						
CO4						
CO5						

Course Content

Unit-1: Advanced Machine Learning Algorithms

1.1. Ensemble Methods
   1.1.1. Random Forests
   1.1.2. Gradient Boosting (XGBoost, LightGBM)
1.2. Support Vector Machines
   1.2.1. Concepts of SVM

1.2.2. Kernel Tricks in SVM
1.3. Naive Bayes Techniques
1.3.1. Gaussian Naive Bayes
1.3.2. Multinomial and Bernoulli Naive Bayes
1.4. Feature Engineering & Data Handling
1.4.1. Feature selection and transformation techniques
1.4.2. Handling imbalanced datasets (SMOTE, class weighting

Unit-2: Deep Learning Fundamentals

2.1 Neural Network Basics
2.1.1 Perceptrons and neural architecture
2.1.2 Activation functions
2.1.3 Backpropagation algorithm
2.2 Deep Learning Frameworks
2.2.1 Introduction to TensorFlow
2.2.2 Working with Keras
2.3 Convolutional Neural Networks (CNNs)
2.3.1 CNN architecture
2.3.2 Image classification using CNNs
2.4 Sequential Models
2.4.1 Introduction to RNNs
2.4.2 Long Short-Term Memory (LSTM)
2.5 Transfer Learning
2.5.1 Pre-trained models (VGG, ResNet)
2.5.2 Fine-tuning techniques

Unit-3: Model Evaluation, Optimization & Interpretability

3.1 Model Validation Techniques
3.1.1 Cross-validation strategies
3.2 Hyperparameter Optimization
3.2.1 Grid Search
3.2.2 Random Search
3.2.3 Bayesian Optimization
3.3 Model Performance Concepts
3.3.1 Bias-variance tradeoff
3.3.2 Evaluation metrics for classification and regression
3.4 Model Interpretability
3.4.1 SHAP (SHapley Additive exPlanations)
3.4.2 LIME (Local Interpretable Model-agnostic Explanations)
3.5 Deployment Evaluation
3.5.1 A/B testing for model deployment

Unit-4: MLOps – Deployment & Lifecycle Management

4.1 ML Pipeline Automation
4.1.1 End-to-end ML pipeline concepts
4.2 Model Versioning
4.2.1 Version control with MLflow
4.3 Containerization

	4.3.1 Docker basics for ML models 4.4 Deployment Strategies 4.4.1 Deployment using Flask API 4.4.2 Deployment using FastAPI 4.5 Model Monitoring 4.5.1 Model performance monitoring 4.5.2 Drift detection techniques 4.6 CI/CD for Machine Learning 4.6.1 Automation of ML workflows 4.7 Cloud ML Platforms 4.7.1 AWS SageMaker overview 4.7.2 Azure Machine Learning basics

Reference Books

No	Title	Author(s)	Publisher	ISBN
1	Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow	AurélienGéron	O'Reilly Media	978-1492032649
2	Deep Learning	Ian Goodfellow, YoshuaBengio, Aaron Courville	MIT Press	978-0262035613
3	Deep Learning with Python	François Chollet	Manning	978-1617294433
4	Pattern Recognition and Machine Learning	Christopher M. Bishop	Springer	978-0387310732
5	Machine Learning: A Probabilistic Perspective	Kevin P. Murphy	MIT Press	978-0262018029
6	Designing Machine Learning Systems	Chip Huyen	O'Reilly Media	978-1098107963
7	Machine Learning Engineering	Andriy Burkov	True Positive Inc.	978-1999579500
8	Practical MLOps	Noah Gift, Alfredo Deza	O'Reilly Media	978-1098103019
9	Introducing MLOps	Mark Treveil, Nicolas Omont, Clément Stenac	O'Reilly Media	978-1492083290
10	Building Machine Learning Powered Applications	Emmanuel Ameisen	O'Reilly Media	978-1492045113

Teaching Methodology	Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.
Evaluation Method	Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. 50% Internal assessment - Attendance, class and home assignment, Unit tests.

	- Practical exam, viva-voce, Journal 50% External Assessment. - Written theory exam - Practical exam, Viva-Voce

CS-703-02: Backend Development & Micro services

[Subject code for Thoery-2611001007022003] [Subject code for Practical-2611001007022004]

Program Name	B.Sc. Computer Science (Honours)					
Semester	7					
NCrf Credit Level	6 Cr					
Course Type	Major					
Course Subtype	Skill Development					
Subject Type	Discipline Specific					
Course Code	703-02					
Course Level	400-499					
Course Title	Backend Development & Micro services					
Credit	Theory	02 Hr	Practical	04 Hr	Total	04 Cr
Effective From	Academic Year: 2026-27					

Course outcome

CO1	Apply advanced Node.js and Express.js concepts to design and implement scalable, secure, and efficient backend applications.
CO2	Design relational and NoSQL database schemas, optimize queries, and integrate ORMs such as Sequelize for effective data management.
CO3	Architect and deploy microservices solutions using service decomposition, API gateways, inter-service communication, and event-driven frameworks.
CO4	Implement robust security practices including authentication, authorization, input validation, and protection against common web vulnerabilities.
CO5	Optimize backend performance through caching strategies, profiling, monitoring, and database query optimization for real-world applications.

Course Outcomes (COs) and Mapping with PSOs

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6
CO1						
CO2						
CO3						
CO4						
CO5						

Course Content

Unit–1: Advanced Node.js & Express.js

1.1 Asynchronous Programming Patterns

1.2 Event Driven Architecture

1.3 Building RESTful APIs with Express.js
1.4 Middleware Implementation
1.5 Error Handling and Logging Strategies
1.6 Authentication & Authorization
  1.6.1 JWT
  1.6.2 OAuth 2.0
  1.6.3 Session Management

Unit–2: Database Design & ORM

2.1 Relational Database Design Principles
2.2 PostgreSQL Advanced Features
  2.2.1 Indexing
  2.2.2 Views
  2.2.3 Stored Procedures
2.3 Sequelize ORM for Node.js
2.4 MongoDB Advanced Queries and Aggregation
2.5 Database Transactions and ACID Properties
2.6 Connection Pooling and Query Optimization

Unit–3: Microservices Architecture

3.1 Monolithic vs Microservices Comparison
3.2 Service Decomposition Strategies
3.3 Inter-Service Communication
  3.3.1 REST
  3.3.2 gRPC
  3.3.3 Message Queues
3.4 API Gateway Pattern
3.5 Service Discovery and Load Balancing
3.6 Event Driven Architecture Introduction
  3.6.1 RabbitMQ
  3.6.2 Kafka

Unit–4: Security & Performance

4.1 API Security Best Practices
  4.1.1 CORS
  4.1.2 Rate Limiting
  4.1.3 Input Validation
4.2 SQL Injection and XSS Prevention
4.3 HTTPS and SSL/TLS Implementation
4.4 Caching Strategies
  4.4.1 Redis Implementation
4.5 Performance Monitoring and Profiling
4.6 Database Query Optimization Techniques

Reference Books

No.	Book Title	Author(s)	Publisher	ISBN
1	Node.js Design Patterns (3rd Edition)	Mario Casciaro, Luciano Mammino	Packt Publishing	978-1839214110

	2	Express in Action: Writing, building, and testing Node.js applications	Evan Hahn	Manning Publications	978-1617292422
	3	Learning Node.js Development	Andrew Mead	Packt Publishing	978-1788626857
	4	Mastering PostgreSQL in Application Development	Dimitri Fontaine	Self-published	978-2956444700
	5	MongoDB: The Definitive Guide (3rd Edition)	Shannon Bradshaw, Eoin Brazil, Kristina Chodorow	O'Reilly Media	978-1491954462
	6	Sequelize ORM Essentials	Jason Brown	Packt Publishing	978-1783983047
	7	Microservices Patterns: With examples in Java	Chris Richardson	Manning Publications	978-1617294549
	8	Building Microservices (2nd Edition)	Sam Newman	O'Reilly Media	978-1492034026
	9	Designing Data-Intensive Applications	Martin Kleppmann	O'Reilly Media	978-1449373320
	10	Web Application Security: Exploitation and Countermeasures for Modern Web Applications	Andrew Hoffman	O'Reilly Media	978-1492053119

Teaching Methodology	Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.
Evaluation Method	Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. 50% Internal assessment - Attendance, class and home assignment, Unit tests. - Practical exam, viva-voce, Journal 50% External Assessment. - Written theory exam - Practical exam, Viva-Voce

704-01 : Natural Language Processing & Computer Vision

[Subject code for Theory-2611001007033001] [Subject code for Practical-2611001007033002]

Program Name	B.Sc. Computer Science (Honours)					
Semester	7					
NCrf Credit Level	6					
Course Type	Major					
Course Subtype	Skill Development					
Subject Type	Discipline Specific					
Course Code	704-01					
Course Level	400-499					
Course Title	Natural Language Processing & Computer Vision					
Credit	Theory	02 Hr	Practical	04 Hr	Total	04 Cr
Effective From	Academic Year: 2026-27					
Course outcome	CO1 Understand fundamental concepts of Natural Language Processing and text preprocessing techniques. CO2 Apply machine learning and deep learning methods for text classification, sentiment analysis, and NER tasks. CO3 Analyze and implement transformer-based models (BERT, GPT) for advanced NLP applications. CO4 Develop computer vision models for image processing, object detection, and segmentation using CNNs and OpenCV. CO5 Design real-world AI applications integrating NLP and Computer Vision techniques for practical use cases.					

Course Outcomes (COs) and Mapping with PSOs

COs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1						
CO2						
CO3						
CO4						
CO5						

Course Content

Unit-1: Natural Language Processing Fundamentals

1.1 Text Preprocessing

	1.1.1 Tokenization 1.1.2 Stemming 1.1.3 Lemmatization 1.1.4 Text Cleaning Techniques **1.2 Text Representation** 1.2.1 Bag of Words (BoW) 1.2.2 Term Frequency (TF) 1.2.3 Inverse Document Frequency (IDF) 1.2.4 TF-IDF Vectorization **1.3 Word Embeddings** 1.3.1 Introduction to Word Embeddings 1.3.2 Word2Vec (CBOW, Skip-gram) 1.3.3 GloVe Model 1.3.4 Comparison of Embedding Techniques **1.4 NLP Applications** 1.4.1 Sentiment Analysis Implementation 1.4.2 Named Entity Recognition (NER) 1.4.3 POS Tagging Basics **1.5 Text Classification** 1.5.1 Traditional ML Methods (Naive Bayes, SVM) 1.5.2 Deep Learning Methods (RNN, LSTM) 1.5.3 Model Evaluation Metrics **Unit-2: Advanced NLP with Transformers** **2.1 Sequence Models** 2.1.1 Sequence-to-Sequence Models 2.1.2 Encoder-Decoder Architecture 2.1.3 Limitations of Traditional Models **2.2 Attention Mechanism** 2.2.1 Concept of Attention 2.2.2 Self-Attention Mechanism 2.2.3 Importance in NLP **2.3 Transformer Architecture** 2.3.1 Encoder Structure 2.3.2 Decoder Structure 2.3.3 Multi-head Attention 2.3.4 Positional Encoding **2.4 Pre-trained Models** 2.4.1 BERT Model Overview 2.4.2 GPT Model Overview 2.4.3 Differences between BERT and GPT **2.5 Hugging Face Implementation** 2.5.1 Transformers Library Introduction 2.5.2 Loading Pre-trained Models 2.5.3 Fine-tuning Techniques

	2.6 NLP Applications with Transformers 2.6.1 Text Generation 2.6.2 Text Summarization 2.6.3 Question Answering Systems **Unit-3: Computer Vision Fundamentals** **3.1 Image Processing Basics** 3.1.1 Image Representation (Pixels, Channels) 3.1.2 Filters (Blurring, Sharpening) 3.1.3 Edge Detection (Canny, Sobel) 3.1.4 Morphological Operations **3.2 Feature Extraction** 3.2.1 SIFT (Scale-Invariant Feature Transform) 3.2.2 SURF (Speeded-Up Robust Features) 3.2.3 HOG (Histogram of Oriented Gradients) **3.3 Image Classification** 3.3.1 Introduction to CNN 3.3.2 Convolution Layer 3.3.3 Pooling Layer 3.3.4 Fully Connected Layer **3.4 Object Detection** 3.4.1 YOLO Algorithm 3.4.2 R-CNN Family (R-CNN, Fast, Faster) 3.4.3 Bounding Box Concepts **3.5 Image Segmentation** 3.5.1 Semantic Segmentation 3.5.2 Instance Segmentation 3.5.3 Mask-based Techniques **3.6 OpenCV Practical Implementation** 3.6.1 Image Input/Output 3.6.2 Image Transformations 3.6.3 Drawing and Visualization 3.6.4 Real-time Camera Processing **Unit-4: Advanced Computer Vision & Applications** **4.1 Transfer Learning** 4.1.1 Concept of Transfer Learning 4.1.2 Pre-trained Models (VGG, ResNet) 4.1.3 Fine-tuning Techniques **4.2 Data Augmentation** 4.2.1 Rotation 4.2.2 Flipping 4.2.3 Scaling 4.2.4 Noise Addition **4.3 Facial Recognition Systems**

4.3.1 Face Detection
4.3.2 Face Recognition Techniques
4.3.3 Har Cascade / Deep Learning Methods
4.4 Optical Character Recognition (OCR)
4.4.1 Text Detection
4.4.2 Text Extraction
4.4.3 OCR Tools (Tesseract)
4.5 Real-time Video Processing
4.5.1 Video Capture using OpenCV
4.5.2 Frame Processing
4.5.3 Object Tracking
4.6 Applications of Computer Vision
4.6.1 Medical Imaging
4.6.2 Autonomous Vehicles
4.6.3 Security Systems
4.7 Integration & Deployment
4.7.1 Web Integration (Flask/Django)
4.7.2 Mobile Integration (APIs)
4.7.3 Model Deployment Basics

Reference Books

No.	Title	Author(s)	Publisher	ISBN
1	Practical Natural Language Processing	Sowmya Vajjala et al.	O'Reilly Media	978-1492054054
2	Speech and Language Processing	Daniel Jurafsky, James H. Martin	Pearson	978-0131873216
3	Natural Language Processing with Python	Steven Bird, Ewan Klein, Edward Loper	O'Reilly Media	978-0596516499
4	Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow	AurélienGéron	O'Reilly Media	978-1492032649
5	Deep Learning	Ian Goodfellow, Yoshua Bengio, Aaron Courville	MIT Press	978-0262035613
6	Computer Vision: Algorithms and Applications	Richard Szeliski	Springer	978-1848829350
7	Learning OpenCV 4	Gary Bradski, Adrian Kaehler	O'Reilly Media	978-1492055747
8	Pattern Recognition and Machine Learning	Christopher M. Bishop	Springer	978-0387310732

	9	Transformers for Natural Language Processing	Denis Rothman	Packt Publishing	978-1800565791
	10	Hands-On Computer Vision with TensorFlow 2	Benjamin Planche, Eliot Andres	Packt Publishing	978-1788830645
Teaching Methodology	Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.				
Evaluation Method	Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. 50% Internal assessment - Attendance, class and home assignment, Unit tests. - Practical exam, viva-voce, Journal 50% External Assessment. - Written theory exam - Practical exam, Viva-Voce				

704 -02 Full Stack Application Development

[Subject code for Theory-2611001007033003] [Subject code for Practical-2611001007033004]

Program Name	B.Sc. Computer Science (Honours)
Semester	7
NCrf Credit Level	6
Course Type	Major
Course Subtype	Skill Development
Subject Type	Discipline Specific
Course Code	704-02
Course Level	400-499
Course Title	Full Stack Application Development

Credit	Theory	02 Hr	Practical	04 Hr	Total	04 Cr

Effective From	Academic Year: 2026-27
Course outcome	CO1 -Understand and build UI using React components and hooks CO2-Develop and validate forms with event handling CO3-Design REST APIs and integrate frontend with backend CO4-Implement routing and state management techniques CO5-Build and deploy a complete full stack web application

Course Outcomes (COs) and Mapping with PSOs

COs	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**
CO1						
CO2						
CO3						
CO4						
CO5						

Course Content

Unit – 1: Advanced React Concepts and Performance Optimization

1.1 Advanced Component Architecture

1.1.1 Higher Order Components (HOC)

1.1.2 Render Props Pattern

1.1.3 Controlled vs Uncontrolled advanced scenarios

1.2 Performance Optimization Techniques

1.2.1 Memoization using React.memo

1.2.2 useMemo and useCallback hooks

1.2.3 Lazy loading with React.lazy and Suspense

	1.3 Advanced State Handling 1.3.1 State colocation and optimization 1.3.2 Avoiding unnecessary re-renders 1.3.3 Immutable state management **1.4 Error Handling in React** 1.4.1 Error Boundaries 1.4.2 Debugging React applications **1.5 Styling and UI Design** 1.5.1 CSS Modules 1.5.2 Styled Components 1.5.3 UI frameworks (Material UI / Bootstrap) **Unit – 2: Backend Development and API Design** **2.1 Advanced Node.js Concepts** 2.1.1 Event loop and asynchronous programming 2.1.2 Middleware architecture **2.2 Express.js Advanced Usage** 2.2.1 Routing and modular structure 2.2.2 Middleware implementation 2.2.3 Error handling in Express **2.3 REST API Design Principles** 2.3.1 API structuring and best practices 2.3.2 Status codes and error responses 2.3.3 API testing using Postman **2.4 Authentication and Authorization** 2.4.1 JWT-based authentication 2.4.2 Role-based access control **2.5 API Integration with React** 2.5.1 Axios interceptors 2.5.2 Handling async operations and errors **Unit – 3: Database Design and Full Stack Integration** **3.1 Advanced MongoDB Concepts** 3.1.1 Schema design principles 3.1.2 Indexing and performance tuning 3.1.3 Aggregation framework **3.2 Mongoose Advanced Operations** 3.2.1 Schema validation 3.2.2 Relationships and population 3.2.3 Middleware in Mongoose **3.3 Full Stack Architecture** 3.3.1 MVC architecture 3.3.2 Separation of concerns 3.3.3 API layer structuring **3.4 Real-time Data Handling** 3.4.1 Introduction to WebSockets 3.4.2 Real-time communication using Socket.io **3.5 Integration Workflow** 3.5.1 End-to-end data flow (Frontend → Backend → Database) 3.5.2 Error handling across layers **Unit – 4: Advanced State Management, Deployment and DevOps Practices** **4.1 Advanced State Management** 4.1.1 Redux Toolkit

4.1.2 Middleware (Thunk, Saga overview)
4.1.3 Global state design patterns

4.2 Advanced React Hooks Usage

4.2.1 Custom hook design patterns
4.2.2 Reusability and abstraction
4.2.3 Hook performance optimization

4.3 Testing and Debugging

4.3.1 Unit testing using Jest
4.3.2 Component testing
4.3.3 Debugging tools

4.4 Deployment and CI/CD

4.4.1 Frontend deployment strategies
4.4.2 Backend deployment and environment configuration
4.4.3 CI/CD pipeline basics

4.5 Version Control and Collaboration

4.5.1 Git advanced commands
4.5.2 Branching strategies
4.5.3 Collaboration using GitHub

Reference Books

No.	Title	Author(s)	Publisher	ISBN
1	Learning React	Alex Banks, Eve Porcello	O'Reilly Media	9781492051725
2	React Explained	Zac Gordon	Independently Published	9781792047848
3	Full Stack React Projects	Shama Hoque	Packt Publishing	9781788835534
4	MERN Stack Web Development	Vasan Subramanian	Packt Publishing	9781787281080
5	Pro MERN Stack	Vasan Subramanian	Apress	9781484230039
6	Node.js, Express, MongoDB & More	Jonas Schmedtmann	Independently Published	9781492050431
7	Web Development with Node and Express	Ethan Brown	O'Reilly Media	9781492053514
8	MongoDB: The Definitive Guide	Shannon Bradshaw, et al.	O'Reilly Media	9781491954461
9	Learning Redux	Daniel Bugl	Packt Publishing	9781789342093
10	Git and GitHub Guide	Ravinder Deol	BPB Publications	9789388511827

Teaching Methodology

Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.

Evaluation Method

Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks.
External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks.

	50% Internal assessment - Attendance, class and home assignment, Unit tests. - Practical exam, viva-voce, Journal 50% External Assessment. - Written theory exam - Practical exam, Viva-Voce

705 Project

<table>
<tr><td>Program Name</td><td>B.Sc. (Computer Science)</td></tr>
<tr><td>Semester</td><td>07</td></tr>
<tr><td>NCRF Credit Level</td><td>6</td></tr>
<tr><td>Course Type</td><td>Major</td></tr>
<tr><td>Course SubType</td><td>Subject specific</td></tr>
<tr><td>Subject Type</td><td>Skill Development</td></tr>
<tr><td>Course Code</td><td>705</td></tr>
<tr><td>Course Level</td><td>400-499</td></tr>
<tr><td>Course Title</td><td>Project</td></tr>
<tr><td>Credits</td><td>6 Cr</td></tr>
<tr><td>Effective From:</td><td>A.Y. 2026-2027</td></tr>
<tr><td>Course Outcomes:</td><td>CO1: Identify and define a real-world industry problem and propose a feasible project plan.
CO2: Apply appropriate methodologies, tools, and technologies to implement the project solution.
CO3: Collaborate effectively with industry mentors and project team members through agile practices.
CO4: Develop and demonstrate a working solution with technical documentation and code repository.
CO5: Present the project outcome through a well-structured report, deployment, and viva-voce.</td></tr>
<tr><td>Mapping Between COs and PSOs.</td><td>
<table>
<tr><th></th><th>PSO 1</th><th>PSO 2</th><th>PSO 3</th><th>PSO 4</th><th>PSO 5</th><th>PSO 6</th><th>PSO 7</th><th>PSO 8</th></tr>
<tr><td>CO1</td><td>■</td><td></td><td></td><td></td><td>■</td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td>■</td><td></td><td></td><td>■</td><td>■</td><td></td><td>■</td><td>■</td></tr>
<tr><td>CO3</td><td></td><td></td><td>■</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td>■</td><td>■</td><td></td><td>■</td><td></td><td>■</td><td></td><td></td></tr>
<tr><td>CO5</td><td>■</td><td>■</td><td></td><td>■</td><td>■</td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Course Content</td><td>This course aims to provide students with hands-on, full-time industry project experience where they apply, integrate, and demonstrate knowledge and skills acquired from previous coursework including software development, data science, AI, cloud deployment, version control, web technologies, and visualization tools. The project will help students learn project planning, communication with industry mentors, adherence to professional practices, and delivery of a functional, demonstrable solution. Emphasis is placed on real-world problem-solving, client interaction, teamwork, version control, documentation, and deployment. It is desirable that students join an organization (through Industry / Software firms / Any other relevant organization / internship with any software development organization) in process to apply their acquired knowledge related to their selected Track. Students will develop and deploy a reasonable size of project applying all relevant skills and understand the project development phases. Project work will be on any of the platforms including Web technology, Android and mobile technology with integration.</td></tr>
</table>

Teaching Methodology	Class work, Discussion, Self-Study, Assignment, Homework, Activity , Self- Assessment etc.
Evaluation Method	50% Internal : Based on presentation of the project and viva-voce. 50% External: Based on project presentation and viva-voce. **Component Weightage** ● Problem Definition & Proposal 10% ● Design and Implementation 30% ● Testing & Deployment 20% ● Project Report 15% ● Presentation & Viva 25% The evaluation of the external assessment will be carried out by panel of three examiners: (i) One examiner from the same institute (ii) One examiner from software industry/corporate (iii) One examiner from other institute affiliated to the university having minimum 11 years of teaching experience at graduation level in computer faculty.

801- Artificial Intelligence & Intelligent Systems

Program Name	B.Sc. Computer Science (Honours)
Semester	8
NCrf Credit Level	6
Course Type	Minor
Course Subtype	Skill Development
Subject Type	Discipline Specific
Course Code	801
Course Level	300-399
Course Title	Artificial Intelligence & Intelligent Systems
Credit	4 Cr
Effective From	Academic Year: 2026-27
Pre- requisite	Knowledge of Python, RDBMS, statistical methods, and basic cloud computing concepts
Course outcome	CO1-Understand fundamental concepts of Artificial Intelligence and intelligent agent design. CO2-Apply search algorithms and problem-solving techniques to AI-based applications. CO3-Represent knowledge using logical models and implement reasoning mechanisms. CO4-Develop basic NLP and intelligent systems using tools like NLTK. CO5- Analyze modern AI applications, including recommendation systems and reinforcement learning, with ethical considerations.

Course Outcomes (COs) and Mapping with PSOs

COs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	
CO1		✓				✓	
CO2	✓		✓	✓			
CO3		✓	✓		✓		✓
CO4	✓			✓	✓	✓	✓
CO5		✓	✓	✓	✓	✓	

Course Content

Unit – 1: Foundations of Artificial Intelligence

1.1 Introduction to Artificial Intelligence
1.2 Intelligent Agents and Environments
1.3 Problem Solving using Search Techniques
1.4 Informed vs Uninformed Search
1.5 Heuristic Functions

Unit – 2: Knowledge Representation and Reasoning

2.1 Knowledge Representation Techniques
2.2 Propositional Logic and Predicate Logic
2.3 Inference Mechanisms
2.4 Rule-Based Systems
2.5 Expert Systems

Unit – 3: Natural Language Processing and AI Applications

3.1 Introduction to NLP
3.2 Text Preprocessing Techniques
3.3 Sentiment Analysis
3.4 Chatbots and Virtual Assistants
3.5 Applications using NLTK

Unit – 4: Intelligent Systems and Emerging Trends

4.1 Reinforcement Learning
4.2 Recommendation Systems
4.3 Computer Vision Basics
4.4 AI in Industry (Healthcare, Smart Cities, Finance)
4.5 Ethics, Bias, and Responsible A

Reference Books

No.	Title	Author(s)	Publisher	ISBN	
1	Artificial Intelligence: A Modern Approach	Stuart Russell, Peter Norvig	Pearson	9780134610993	
2	Artificial Intelligence	Elaine Rich, Kevin Knight	McGraw-Hill	9780070087705	
3	Artificial Intelligence: Foundations of Computational Agents	David Poole, Alan Mackworth	Cambridge	9781107195394	
4	AI: A Guide to Intelligent Systems	Michael Negnevitsky	Pearson	9780329034894	
5	Introduction to Artificial Intelligence	Wolfgang Ertel	Springer	9783319584478	
6	Artificial Intelligence Basics	Tom Taulli	Apress	9781484227657	
7	Machine Learning and AI	Amit Konar	CRC Press	9780367389376	
8	Natural Language Processing with Python	Steven Bird	O'Reilly	9780596516499	
9	Speech and Language Processing	Daniel Jurafsky, James H. Martin	Pearson	9780131873216	

	10	Artificial Intelligence for Humans	Jeff Heaton	Heaton Research	9781508847656	
Teaching Methodology	Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.					
Evaluation Method	Internal Assessment: 50 marks Theory = 50 Marks. External Assessment: 50 marks Theory = 50 Marks. 50% Internal assessment - Attendance, class and home assignment, Unit tests. 50% External Assessment. - Written theory exam					

CS-802: Cloud Native Application Development

[Subject code for Theory-2711001008011001] [Subject code for Practical-02711001008011002]

Program Name	B.Sc. Computer Science (Honors)
Semester	8
NCrf Credit Level	4 Cr
Course Type	Major
Course Subtype	Skill Development
Subject Type	Discipline Specific
Course Code	802
Course Level	400-499
Course Title	Cloud Native Application Development

Credit	Theory	02 Hr	Practical	04 Hr	Total	04 Cr

Effective From	Academic Year: 2026-27

Course outcome

CO1	Understand serverless computing and cloud-native architecture
CO2	Gain practical knowledge of cloud database services
CO3	Implement secure cloud applications using IAM and encryption
CO4	Monitor and optimize cloud resources effectively
CO5	Deploy and manage applications using Infrastructure as Code (IaC)

Course Outcomes (COs) and Mapping with PSOs

COs	**PSO1**	**PSO2**	**PSO3**	**PSO4**	**PSO5**	**PSO6**
CO1						
CO2						
CO3						
CO4						
CO5						

Course Content

Unit-1: Server less Architecture & Functions

1.1 Serverless Computing Fundamentals

1.1.1 Serverless Computing Paradigm
1.1.2 Benefits and Limitations of Serverless Architecture
1.1.3 Event-driven Architecture Concepts

	1.2 Cloud Functions Platforms 1.2.1 AWS Lambda Fundamentals 1.2.2 Azure Functions 1.2.3 Google Cloud Functions **1.3 Function Execution & Optimization** 1.3.1 Function Triggers and Events 1.3.2 Cold Start Problem and Optimization Techniques 1.3.3 Performance Tuning Strategies **1.4 Serverless Deployment Tools** 1.4.1 Serverless Framework 1.4.2 AWS SAM (Serverless Application Model) 1.4.3 Deployment and Version Management **1.5 Cost Optimization** 1.5.1 Pay-per-use Pricing Model 1.5.2 Resource Optimization Techniques 1.5.3 Monitoring Cost Efficiency **Unit-2: Cloud Database Services** **2.1 Cloud-native Databases** 2.1.1 AWS RDS 2.1.2 Amazon Aurora 2.1.3 DynamoDB **2.2 Multi-Cloud Database Services** 2.2.1 Azure Cosmos DB 2.2.2 Google Cloud Firestore 2.2.3 Comparison of Cloud Databases **2.3 Database Migration & Management** 2.3.1 Migration Strategies to Cloud 2.3.2 Data Synchronization Techniques 2.3.3 Backup and Disaster Recovery **2.4 Scalability & Deployment** 2.4.1 Multi-region Deployment 2.4.2 Database Scaling (Vertical & Horizontal) 2.4.3 High Availability Concepts **2.5 Database Selection** 2.5.1 Choosing the Right Database Service 2.5.2 SQL vs NoSQL Databases 2.5.3 Use-case Based Selection **Unit-3: Cloud Security & Identity Management** **3.1 Identity and Access Management (IAM)** 3.1.1 IAM Policies and Roles 3.1.2 User and Permission Management 3.1.3 Principle of Least Privilege **3.2 Authentication Services** 3.2.1 AWS Cognito for User Authentication 3.2.2 Azure Active Directory 3.2.3 Identity Federation Concepts **3.3 Secret Management**

3.3.1 AWS Secrets Manager
3.3.2 Azure Key Vault
3.3.3 Secure Credential Storage
3.4 Data Security
3.4.1 Encryption at Rest
3.4.2 Encryption in Transit
3.4.3 Key Management Services
3.5 Network Security
3.5.1 Virtual Private Cloud (VPC)
3.5.2 Security Groups and Firewalls
3.5.3 Network Access Control
3.6 Compliance & Auditing
3.6.1 Security Compliance Standards
3.6.2 Logging and Auditing
3.6.3 Risk Management

Unit-4: Monitoring, Logging & Cost Optimization
4.1 Monitoring Tools
4.1.1 AWS CloudWatch
4.1.2 Azure Monitor
4.1.3 Google Cloud Logging
4.2 Performance Monitoring
4.2.1 Application Performance Monitoring (APM)
4.2.2 Distributed Tracing
4.2.3 Performance Metrics Analysis
4.3 Logging & Analysis
4.3.1 Log Aggregation
4.3.2 Log Analysis Techniques
4.3.3 Debugging and Issue Tracking
4.4 Cost Optimization Techniques
4.4.1 Cost Monitoring and Budgeting
4.4.2 Right-sizing Resources
4.4.3 Reserved Instances and Spot Instances
4.5 Infrastructure as Code (IaC)
4.5.1 Introduction to Terraform
4.5.2 AWS CloudFormation Templates
4.5.3 Automation of Infrastructure Deployment

Reference Books

No.	Title	Author(s)	Publisher	ISBN	
1	Cloud Computing: Concepts, Technology & Architecture	Thomas Erl, Ricardo Puttini, Zaigham Mahmood	Pearson	978-0133387520	
2	AWS Certified Solutions Architect Official Study Guide	Ben Piper, David Clinton	Sybex (Wiley)	978-1119504214	

	3	Serverless Architectures on AWS	Peter Sbarski	Manning Publications	978-1617293825	
	4	Architecting the Cloud: Design Decisions for Cloud Computing Service Models	Michael J. Kavis	Wiley	978-1118447826	
	5	Cloud Native Patterns	Cornelia Davis	Manning Publications	978-1617294297	
	6	Learning AWS: Design, Build, and Deploy Responsive Applications	Aurobindo Sarkar, Amit Shah	Packt Publishing	978-1787281066	
	7	AWS Lambda in Action	Danilo Poccia	Manning Publications	978-1617293719	
	8	Mastering Azure Serverless Computing	Jamie Maguire	Packt Publishing	978-1788622295	
	9	Google Cloud Platform in Action	JJ Geewax	Manning Publications	978-1617293528	
	10	Terraform: Up & Running	Yevgeniy Brikman	O'Reilly Media	978-1492046905	
Teaching Methodology	Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.					
Evaluation Method	Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. 50% Internal assessment - Attendance, class and home assignment, Unit tests. - Practical exam, viva-voce, Journal 50% External Assessment. - Written theory exam - Practical exam, Viva-Voce					

Sample Practical List

Unit–1: Serverless Architecture & Functions

Practical 1: Setup cloud account and deploy a basic function using AWS Lambda / Azure Functions *(CO1, CO7)*
Practical 2: Create event-driven function triggered by HTTP request (API Gateway) *(CO1, CO4)*
Practical 3: Implement file upload trigger (e.g., S3 / Blob Storage event trigger) *(CO1, CO4)*
Practical 4: Develop serverless REST API using Lambda + API Gateway *(CO1, CO4, CO7)*
Practical 5: Compare execution time and cold start behavior of functions *(CO2, CO6)*
Practical 6: Use Serverless Framework / AWS SAM for deployment automation *(CO1, CO7)*

Unit–2: Cloud Database Services

Practical 7: Setup and connect to cloud database (AWS RDS / Firebase Firestore) *(CO2, CO4)*
Practical 8: Perform CRUD operations using cloud database APIs *(CO2, CO4)*
Practical 9: Implement NoSQL database using DynamoDB / MongoDB Atlas *(CO2, CO4)*
Practical 10: Design and deploy multi-region database setup *(CO2, CO3)*
Practical 11: Perform database backup and recovery simulation *(CO2, CO3)*
Practical 12: Compare SQL vs NoSQL performance for given dataset *(CO2, CO6)*

Unit–3: Cloud Security & Identity Management

Practical 13: Create IAM users, roles, and policies *(CO3)*
Practical 14: Implement authentication using AWS Cognito / Firebase Auth *(CO3, CO4)*
Practical 15: Secure API using token-based authentication *(JWT/OAuth basics)(CO3, CO4)*
Practical 16: Configure VPC and security groups for cloud application *(CO3)*
Practical 17: Implement encryption for data storage and communication *(CO3)*
Practical 18: Manage secrets using AWS Secrets Manager / environment variables *(CO3, CO5)*

Unit–4: Monitoring, Logging & Cost Optimization

Practical 19: Setup monitoring using AWS CloudWatch / Azure Monitor *(CO5)*
Practical 20: Implement logging and analyze logs for debugging *(CO5, CO7)*
Practical 21: Create alerts and notifications for system failures *(CO5)*
Practical 22: Analyze resource usage and cost using billing dashboard *(CO6)*
Practical 23: Implement auto-scaling and load balancing *(CO4, CO6)*
Practical 24: Deploy infrastructure using Terraform / CloudFormation *(CO4, CO7)*
Mini Project *(CO1 to CO8 – All Outcomes Covered)*

Practical 25: Develop and deploy a complete cloud-native application:

- Serverless backend (Lambda / Functions)
- Cloud database integration
- Authentication & security
- Monitoring & logging
- Deployment on cloud

CS-803: Emerging Technologies & Industry Practices

[Subject code for Theory-2711001008022001] [Subject code for Practical-2711001008022002]

Program Name	B.Sc. Computer Science (Honors)					
Semester	8					
NCrf Credit Level	6					
Course Type	Major					
Course Subtype	Skill Development					
Subject Type	Discipline Specific					
Course Code	803					
Course Level	400-499					
Course Title	Emerging Technologies & Industry Practices					
Credit	Theory	02 Hr	Practical	04 Hr	Total	04 Cr
Effective From	Academic Year: 2026-27					
Course outcome	**CO1:** Apply emerging technologies in real-world scenarios **CO2:** Analyze use cases of block-chain, IoT, and quantum systems **CO3:** Design innovative and scalable solutions **CO4:** Develop basic decentralized and IoT-based applications **CO5:** Evaluate modern software engineering practices **CO6:** Integrate interdisciplinary technologies in projects **CO7:** Build future-ready applications **CO8:** Demonstrate readiness for evolving IT industry roles					

Course Outcomes (COs) and Mapping with PSOs

	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	✓	✓	–	✓	✓	–
CO2	✓	✓	✓	✓	✓	–
CO3	✓	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓	–
CO5	✓	✓	✓	✓	✓	✓
CO6	✓	✓	✓	✓	✓	–
CO7	✓	✓	✓	✓	✓	✓
CO8	✓	✓	✓	✓	✓	✓

Course Content	**Unit–1: Blockchain & Decentralized Applications** **1.1 Blockchain Fundamentals** 1.1.1 Introduction to Blockchain 1.1.2 Distributed Ledger Technology 1.1.3 Key Features of Blockchain **1.2 Consensus Mechanisms** 1.2.1 Proof of Work (PoW) 1.2.2 Proof of Stake (PoS) 1.2.3 Other Consensus Methods **1.3 Ethereum Platform** 1.3.1 Introduction to Ethereum 1.3.2 Ethereum Architecture **1.4 Smart Contracts & Solidity** 1.4.1 Concept of Smart Contracts 1.4.2 Basics of Solidity Programming **1.5 Web3.js Integration** 1.5.1 Introduction to Web3.js 1.5.2 Interaction with Blockchain **1.6 Cryptocurrencies & Digital Assets** 1.6.1 Cryptocurrency Basics 1.6.2 Tokens and NFTs **1.7 Blockchain Applications** 1.7.1 Real-world Use Cases 1.7.2 Basic DApp Development **Unit–2: Internet of Things (IoT) Advanced Applications** **2.1 IoT Architecture** 2.1.1 Edge Computing 2.1.2 Fog Computing 2.1.3 Cloud Integration **2.2 Communication Protocols** 2.2.1 MQTT Protocol 2.2.2 Other IoT Protocols **2.3 IoT Platforms** 2.3.1 AWS IoT 2.3.2 Azure IoT Hub **2.4 Sensor Data Processing** 2.4.1 Data Collection 2.4.2 Data Processing Techniques **2.5 Hardware Integration** 2.5.1 Raspberry Pi 2.5.2 Arduino Integration **2.6 Real-Time Data Streaming** 2.6.1 Streaming Concepts 2.6.2 Applications in IoT **2.7 IoT Security & Applications** 2.7.1 Security Challenges in IoT 2.7.2 Smart Applications

	Unit–3: Quantum Computing & Future Technologies **3.1 Quantum Computing Basics** 3.1.1 Introduction to Quantum Computing 3.1.2 Classical vs Quantum Computing **3.2 Qubits & Quantum Gates** 3.2.1 Qubits Concept 3.2.2 Quantum Gates **3.3 Quantum Algorithms** 3.3.1 Overview of Algorithms 3.3.2 Applications **3.4 Qiskit Framework** 3.4.1 Introduction to Qiskit 3.4.2 Basic Implementation **3.5 Applications of Quantum Computing** 3.5.1 Cryptography 3.5.2 Machine Learning **3.6 Emerging Technologies** 3.6.1 Neuromorphic Computing Basics 3.6.2 Extended Reality (AR, VR, MR) 3.6.3 Frameworks and Use Cases **Unit–4: Software Engineering Best Practices** **4.1 Design Patterns** 4.1.1 Singleton Pattern 4.1.2 Factory Pattern 4.1.3 Observer Pattern 4.1.4 MVC Pattern 4.1.5 MVVM Pattern **4.2 SOLID Principles** 4.2.1 Overview of SOLID 4.2.2 Application in Development **4.3 Clean Code Practices** 4.3.1 Coding Standards 4.3.2 Maintainability **4.4 Code Reviews & Technical Debt** 4.4.1 Code Review Process 4.4.2 Managing Technical Debt **4.5 Agile & Scrum Methodologies** 4.5.1 Agile Principles 4.5.2 Scrum Framework 4.5.3 Sprint Planning **4.6 Documentation & Open Source** 4.6.1 Documentation Standards 4.6.2 Open-source Contribution
Reference Books	1. Mastering Blockchain, Imran Bashir, Packt Publishing, 978-1839213199 2. Blockchain Basics, Daniel Drescher, Apress, 978-1484226032 3. Internet of Things: A Hands-On Approach, ArshdeepBahga; Vijay Madisetti, VPT, 978-0996025515 4. Designing the Internet of Things, Adrian McEwen; Hakim Cassimally, Wiley,

	978-1118430620 5. Quantum Computing for Everyone, Chris Bernhardt, MIT Press, 978-0262539531 6. Programming Quantum Computers, Eric R. Johnston et al., O'Reilly Media, 978-1492039686 7. Learning Qiskit, Sarah Kaiser, Packt Publishing, 978-1838829520 8. Clean Code, Robert C. Martin, Prentice Hall, 978-0132350884 9. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma et al., Addison-Wesley, 978-0201633610 10. Agile Estimating and Planning, Mike Cohn, Prentice Hall, 978-0131479418
Teaching Methodology	Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.
Evaluation Method	Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks. 50% Internal assessment - Attendance, class and home assignment, Unit tests. - Practical exam, viva-voce, Journal 50% External Assessment. - Written theory exam - Practical exam, Viva-Voce

Course: CS-803: Emerging Technologies & Industry Practices

Sample Practical List

Unit–1: Blockchain & Decentralized Applications

Practical 1: Study and demonstrate basic blockchain working using online simulator *(CO1, CO2)*
Practical 2: Create simple cryptocurrency transaction flow diagram *(CO2, CO3)*
Practical 3: Write and deploy a basic smart contract (Hello World) using Solidity *(CO4)*
Practical 4: Connect smart contract using Web3.js (basic interaction) *(CO4, CO6)*
Practical 5: Explore Ethereum wallet and perform test transaction *(CO1, CO2)*
Practical 6: Identify and present real-world blockchain use case *(CO2, CO8)*

Unit–2: Internet of Things (IoT) Advanced Applications

Practical 7: Draw IoT architecture diagram (edge, fog, cloud) *(CO2, CO3)*
Practical 8: Simulate sensor data using online IoT simulator *(CO1, CO4)*
Practical 9: Basic Arduino/Raspberry Pi LED control program *(CO4)*
Practical 10: Send sensor data to cloud using MQTT protocol *(CO4, CO6)*
Practical 11: Create simple dashboard to display IoT data *(CO3, CO7)*
Practical 12: Study IoT security challenges and write short report *(CO5, CO8)*

Unit–3: Quantum Computing & Future Technologies

Practical 13: Demonstrate basic quantum circuit using Qiskit simulator *(CO1, CO2)*
Practical 14: Create simple quantum gate circuit (Hadamard, NOT gate) *(CO2, CO3)*
Practical 15: Study application of quantum computing in cryptography *(CO2, CO8)*
Practical 16: Demonstrate basic AR/VR application using any tool/platform *(CO1, CO7)*
Practical 17: Explore use cases of XR (AR/VR/MR) in industry *(CO2, CO8)*
Practical 18: Short presentation on future technologies (AI + Quantum + XR) *(CO6, CO8)*

Unit–4: Software Engineering Best Practices

Practical 19: Implement basic design pattern (Singleton / Factory) *(CO3, CO5)*
Practical 20: Refactor code using clean code principles *(CO5)*
Practical 21: Conduct peer code review and prepare report *(CO5, CO6)*
Practical 22: Prepare sprint plan using Agile methodology *(CO5, CO8)*
Practical 23: Create project documentation (SRS or API doc) *(CO5, CO8)*
Practical 24: Contribute to open-source project (minor change or issue) *(CO6, CO8)*
Mini Project *(CO1 to CO8 – Covered at Beginner Level)*
Practical 25: Develop a basic interdisciplinary application:
Example: IoT + Dashboard OR Blockchain + Simple App OR AR Demo

804-01 Data Science and Advanced Analytics

[Subject code for Theory-2711001008033001] [Subject code for Practical-2711001008033002]

Program Name	B.Sc. Computer Science (Honours)
Semester	8
NCrf Credit Level	6
Course Type	Major
Course Subtype	Skill Development
Subject Type	Discipline Specific
Course Code	804-01
Course Level	400-499
Course Title	Data Science and Advanced Analytics
Credit	4
Effective From	Academic Year: 2026-27
Pre- requisite	Machine Learning using Python (CS-503), Data Visualizations (CS-601), Big Data Technologies (CS-702-A)
Course outcome	**CO1** Apply statistical analysis and hypothesis testing techniques (t-tests, chi-square, ANOVA, regression, ARIMA) for data-driven insights and decision validation. **CO2** Design interactive data visualizations and BI dashboards using Tableau, Power BI, Plotly/Dash, and storytelling practices. **CO3** Implement predictive and forecasting models (Prophet, churn prediction, recommendation systems, clustering, market basket analysis) for business applications. **CO4** Develop and manage big data pipelines, including ETL, data warehousing, Apache Airflow orchestration, and cloud analytics platforms. **CO5** Integrate analytics and data engineering skills to build scalable, high-quality, end-to-end solutions for enterprise decision-making.

Course Outcomes (COs) and Mapping with PSOs

CO No.	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	✓	✓	✓	✓		
CO2		✓	✓	✓	✓	✓
CO3	✓	✓	✓		✓	✓
CO4	✓	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓

Course Content

Unit–1: Statistical Analysis & Hypothesis Testing

1.1 Descriptive vs Inferential Statistics

1.1.1 Characteristics of Descriptive Statistics

1.1.2 Inferential Methods and Applications

1.2 Probability Distributions

1.2.1 Normal Distribution

1.2.2 Binomial Distribution

	1.2.3 Poisson Distribution 1.2.4 Applications in Analytics **1.3 Hypothesis Testing** 1.3.1 One Sample t-test 1.3.2 Two Sample t-test 1.3.3 Chi-square Test 1.3.4 ANOVA (One-way, Two-way) **1.4 Correlation & Regression** 1.4.1 Pearson Correlation 1.4.2 Spearman Correlation 1.4.3 Simple Regression 1.4.4 Multiple Regression **1.5 Time Series Analysis** 1.5.1 Fundamentals of ARIMA Models 1.5.2 Stationarity and Differencing **1.6 Statistical Significance** 1.6.1 p-values 1.6.2 Confidence Intervals 1.6.3 Type I & Type II Errors **Unit–2: Advanced Data Visualization & Business Intelligence** **2.1 Interactive Dashboards** 2.1.1 Plotly Basics 2.1.2 Dash Framework 2.1.3 Building Interactive Charts **2.2 Tableau Advanced Techniques** 2.2.1 Calculated Fields 2.2.2 Parameters 2.2.3 Level of Detail (LOD) Expressions **2.3 Power BI Fundamentals** 2.3.1 Data Modeling 2.3.2 DAX Basics 2.3.3 Report Creation **2.4 Storytelling with Data** 2.4.1 Narrative Techniques 2.4.2 Audience-centric Visualization **2.5 Visualization Best Practices** 2.5.1 Color Theory 2.5.2 Layout Design 2.5.3 Accessibility **2.6 Executive Dashboards** 2.6.1 KPI Tracking 2.6.2 Drill-down Features 2.6.3 Real-time Updates **2.7 Embedded Analytics** 2.7.1 Embedding Dashboards in Applications 2.7.2 API Integration

	Unit–3: Predictive Analytics & Forecasting **3.1 Time Series Forecasting** 3.1.1 Classical Methods 3.1.2 Prophet Library Usage **3.2 Customer Churn Prediction** 3.2.1 Logistic Regression 3.2.2 Random Forest Models **3.3 Recommendation Systems** 3.3.1 Collaborative Filtering 3.3.2 Content-based Filtering **3.4 Market Basket Analysis** 3.4.1 Association Rules 3.4.2 Apriori Algorithm **3.5 Customer Segmentation** 3.5.1 K-means Clustering 3.5.2 Hierarchical Clustering 3.5.3 Profiling Techniques **3.6 Lifetime Value Prediction** 3.6.1 Survival Analysis 3.6.2 Regression-based Approaches **Unit–4: Big Data Analytics & Data Engineering** **4.1 ETL Pipelines** 4.1.1 Extraction 4.1.2 Transformation 4.1.3 Loading 4.1.4 Tools and Frameworks **4.2 Data Warehousing Concepts** 4.2.1 Fact Tables 4.2.2 Dimension Tables 4.2.3 Star Schema Design **4.3 Workflow Orchestration** 4.3.1 Apache Airflow Basics 4.3.2 DAG Creation 4.3.3 Scheduling Workflows **4.4 Data Quality & Validation** 4.4.1 Data Cleaning 4.4.2 Validation Techniques **4.5 Columnar Formats** 4.5.1 Parquet 4.5.2 ORC 4.5.3 Advantages in Big Data **4.6 Data Lakes** 4.6.1 Architecture 4.6.2 Use Cases **4.7 Cloud Analytics Platforms** 4.7.1 AWS Glue 4.7.2 Azure Synapse 4.7.3 Integration with BI Tools

<table>
<tr><td></td><td></td></tr>
<tr><td>Reference Books</td><td>
<table>
<tr><th>No.</th><th>Book Title</th><th>Author(s)</th><th>Publisher</th><th>ISBN</th></tr>
<tr><td>1</td><td>Testing Statistical Hypotheses</td><td>E.L. Lehmann, Joseph P. Romano</td><td>Springer Texts in Statistics</td><td>978-3031218148</td></tr>
<tr><td>2</td><td>Mathematical Statistics with Applications</td><td>Dennis Wackerly, William Mendenhall, Richard L. Scheaffer</td><td>Cengage Learning</td><td>978-0495110811</td></tr>
<tr><td>3</td><td>Storytelling with Data: A Data Visualization Guide for Business Professionals</td><td>Cole NussbaumerKnaflic</td><td>Wiley</td><td>978-1119002253</td></tr>
<tr><td>4</td><td>Microsoft Power BI for Real-World Analytics: A Practical Guide</td><td>Fred Thompson</td><td>Independently Published</td><td>979-8243730907</td></tr>
<tr><td>5</td><td>Forecasting and Predictive Analytics, 7th Edition</td><td>Barry Keating, J. Holton Wilson, Shovan Chowdhury</td><td>McGraw Hill</td><td>978-1260461421</td></tr>
<tr><td>6</td><td>Designing Data-Intensive Applications</td><td>Martin Kleppmann</td><td>O’Reilly Media</td><td>978-1449373320</td></tr>
<tr><td>7</td><td>Handbook of Big Data Analytics</td><td>Wolfgang Karl Härdle, Henry Horng-Shing Lu, et al.</td><td>Springer Handbooks of Computational Statistics</td><td>978-3319182841</td></tr>
<tr><td>8</td><td>Big Data Analytics</td><td>Kamal Raj</td><td>McGraw Hill Education India</td><td>978-9353165138</td></tr>
</table>
</td></tr>
<tr><td>Teaching Methodology</td><td>Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.</td></tr>
<tr><td>Evaluation Method</td><td>Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks.
External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks.
50% Internal assessment
- Attendance, class and home assignment, Unit tests.
- Practical exam, viva-voce, Journal
50% External Assessment.
- Written theory exam
- Practical exam, Viva-Voce</td></tr>
</table>

804-02 Advance Mobile Application Development

[Subject code for Theory-2711001008033003] [Subject code for Practical-2711001008033004]

Program Name	B.Sc. Computer Science (Honours)					
Semester	8					
NCrf Credit Level	6					
Course Type	Major					
Course Subtype	Skill Development					
Subject Type	Discipline Specific					
Course Code	804-02					
Course Level	400-499					
Course Title	Advance Mobile Application Development					
Credit	Theory	02 Hr	Practical	04 Hr	Total	04 Cr
Effective From	Academic Year: 2026-27					
Course outcome	CO1-Analyze and select appropriate cross-platform frameworks for mobile development C02- Design scalable and responsive mobile user interfaces CO3- Implement API integration and database-driven applications with CRUD operations CO4-Apply authentication, state management, and performance optimization techniques CO5-Develop, test, and deploy production-ready mobile applications					

Course Outcomes (COs) and Mapping with PSOs

COs	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1						
CO2						
CO3						
CO4						
CO5						

Course Content

Unit – 1: Cross Platform Mobile Architecture and Development Setup

1.1. Mobile Application Development Paradigms

1.1.1. Native vs Hybrid vs Cross-platform architecture

1.1.2. Performance considerations and use-cases

1.2. Cross Platform Frameworks Overview

1.2.1. Architecture of React Native (Bridge, Components)

1.2.2. Architecture of Flutter (Widgets, Rendering Engine)

1.2.3. Comparison and selection criteria

	1.3. Development Environment Setup 1.3.1. Node.js, npm/yarn, Android Studio, SDK tools 1.3.2. Emulator and debugging tools **1.4. Project Structure and Lifecycle** 1.4.1. Application lifecycle in mobile apps 1.4.2. Folder structure and modular development **1.5. UI/UX Design Principles** 1.5.1. Material Design basics 1.5.2. Responsive and adaptive UI **Unit – 2: Advanced UI Development and Navigation(based on** React Native**)** **2.1 Core Components and Layout System** 2.1.1 View hierarchy, Flexbox layout system 2.1.2 Custom components and reusable UI modules **2.2 Navigation Techniques** 2.2.1 Stack, Tab, Drawer navigation 2.2.2 Navigation lifecycle and state handling **2.3 Event Handling and Gesture System** 2.3.1 Touch handling, gesture recognition **2.4 Forms and Input Handling** 2.4.1 Controlled inputs, validation strategies 2.4.2 Error handling and user feedback **2.5 API Integration** 2.5.1 RESTful services integration 2.5.2 HTTP methods using Fetch/Axios 2.5.3 JSON parsing and error handling **Unit – 3: Database Design and CRUD Operations(based on React Nativ**e) **3.1 Data Management in Mobile Applications** 3.1.1 Structured vs unstructured data 3.1.2 Offline-first vs online-first approach **3.2 Local Database Implementation** 3.2.1 SQLite integration and schema design 3.2.2 Data persistence and transactions **3.3 Cloud Database and Backend Services** 3.3.1 Introduction to Firebase Firestore & Realtime Database 3.3.2 Data synchronization and scalability **3.4 CRUD Operations Implementation** 3.4.1 Create, Read, Update, Delete operations 3.4.2 Handling asynchronous operations **3.5 API & Database Integration** 3.5.1 Connecting mobile app with backend APIs 3.5.2 Data flow and state synchronization **Unit – 4: Advanced Concepts, Security and Deployment** **4.1 State Management** 4.1.1 Context API and advanced hooks

4.1.2 Introduction to Redux (optional integration)

4.2 Authentication and Authorization

4.2.1 User authentication using Firebase/Auth APIs

4.2.2 Token-based authentication (JWT basics)

4.3 Performance Optimization

4.3.1 Lazy loading, memoization

4.3.2 Debugging and profiling tools

4.4 Testing and Debugging

4.4.1 Unit testing basics

4.4.2 Debugging using developer tools

4.5 Build and Deployment

4.5.1 APK/IPA generation

4.5.2 Publishing process (Google Play Store)

4.5.3 Version control using Git

Reference Books

No.	Title	Author(s)	Publisher	ISBN
1	Learning React Native	Bonnie Eisenman	O'Reilly Media	9781491929001
2	React Native in Action	Nader Dabit	Manning Publications	9781617294051
3	Full-Stack React Native	Houssein Djirdeh et al.	Fullstack.io	9780991344628
4	Flutter for Beginners	Alessandro Biessek	Packt Publishing	9781788996082
5	Beginning Flutter	Marco L. Napoli	Wiley	9781119550822
6	Professional Android Development	Reto Meier	Wiley	9781118949528
7	Mobile App Development with Ionic	Chris Griffith	O'Reilly Media	9781491998120
8	RESTful Web APIs	Leonard Richardson, Mike Amundsen	O'Reilly Media	9781449358068
9	Programming Flutter	Carmine Zaccagnino	Pragmatic Bookshelf	9781680506952
10	Firebase Essentials	Rajesh Venkatesan	Packt Publishing	9781788477618

Teaching Methodology

Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions.

Evaluation Method

Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks.
External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks.
50% Internal assessment

- Attendance, class and home assignment, Unit tests.

	- Practical exam, viva-voce, Journal 50% External Assessment. - Written theory exam - Practical exam, Viva-Voce

805: On-the-Job Training and Industry Immersion

Program Name	B.Sc. (Computer Science)
Semester	08
NCRF Credit Level	6 Cr
Course Type	Major
Course SubType	Subject specific
Subject Type	Skill Development
Course Code	805
Course Level	400-499
Course Title	On-the-Job Training and Industry Immersion
Credits	6
Effective From:	A.Y. 2026-2027
Course Outcomes	**CO1**: Understand organizational structure, working hierarchy, and operational workflow in a software company. **CO2**: Identify the technology stack, development practices, and software tools used in real-world environments. **CO3**: Analyze HR functions, team dynamics, and work culture strategies in IT companies. **CO4**: Demonstrate knowledge integration through structured documentation and presentation. **CO5**: Reflect on industry practices and propose self-improvement plans for better employability.
Mapping between Cos and PSOs	

	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5	PSO 6	PSO 7	PSO 8
CO1								
CO2								
CO3								
CO4								
CO5								

Phases and components of Course Outcomes:	**1: Orientation and Preparation for Industry Visit** 1.1 Understanding the Purpose of OJT 1.2 Guidelines for Professional Conduct during Visits 1.3 Basics of Organizational Structures in IT 1.4 Overview of Roles in a Software Company (Developer, QA, Analyst, HR, DevOps, etc.) 1.5 Tools and Templates for Observation, Interview, and Note-Taking **2: Industry Visit and Observation** 2.1 Visiting Software Companies (Startups, Mid-size, or Large IT firms) 2.2 Understanding Software Development Lifecycle (SDLC/Agile) 2.3 Identifying Technologies, Tools, and Platforms in Use 2.4 Observing Team Collaboration, Roles & Responsibilities 2.5 Collecting Data: Hierarchy Charts, Project Flow, Software Architecture

	(as allowed) **3: HR, Strategy, and Organizational Analysis** 3.1 HR Functions: Hiring Strategies, Performance Appraisal, Onboarding 3.2 Employee Benefits and Career Growth Pathways 3.3 Corporate Culture, Diversity, Work-Life Balance Initiatives 3.4 CSR, Sustainability, and Company Vision 3.5 Learning from Employee Interactions: Q&A, Mentorship, Informal Interviews **4: Reporting, Documentation, and Presentation** 4.1 Preparing a Structured Industry Visit Report 4.2 Creating Hierarchy Diagrams, Workflows, Tech Stack Documentation 4.3 Analyzing Key Takeaways and Lessons Learned 4.4 Preparing and Delivering Presentation (Team/Individual) 4.5 Viva-Voce or Reflective Discussion with Faculty Panel
Project Evaluation:	50% Internal : Based on presentation on following components and viva-voce. 50% External: Based on presentation on following components and viva-voce. **Component** **Weightage** Participation & Conduct 10% Observation Logs & Notes 10% Final Report 40% Presentation & Demo 30% Reflective Discussion/Viva 10% The evaluation of the external assessment will be carried out by panel of threeexaminers: (i) One examiner from the same institute (ii) One examiner from software industry/corporate (iii) One examiner from other institute affiliated to the university having minimum 11 years of teaching experience at graduation level in computer faculty..