Re-Accredited B++ 2.86 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 તમામ બી.સી.એ.કોલેજોનાં આચાર્યશ્રીઓને જણાવવાનું કે, શૈક્ષણિક વર્ષ ૨૦૨૬–૨૭ થી અમલમાં આવનાર B.C.A. (Honours) Sem.-7 & 8 નો પેટાસમિતિ દ્વારા તૈયાર કરવામાં આવેલ અભ્યાસક્રમ કોમ્પ્યુટર સાયન્સ વિષયની અભ્યાસ સમિતિની તા.૨૫/૦૩/૨૦૨૬ ની સભાના ઠરાવ ક્રમાંકઃ૧૭ થી મંજૂર કરી કોમ્પ્યુટર સાયન્સ ફેકલ્ટીને કરેલ ભલામણ કોમ્પ્યુટર સાયન્સ ફેકલ્ટીની તા.૨૯/૪/૨૦૨૬ ની સભાના ઠરાવ ક્રમાંકઃ૧૪ થી મંજૂર કરી એકેડેમિક કાઉન્સિલને કરેલ ભલામણ ને એકેડેમિક કાઉન્સિલની તા.૦૭/૦૫/૨૦૨૬ની સભાનાં ઠરાવ ક્રમાંકઃ૫૫ થી મંજૂર કરેલ છે. જેનો અમલ કરવા આથી જાણ કરવામાં આવે છે.

બિડાણઃ ઉપર મુજબ

ક્રમાંકઃઓથો./પરિપત્ર/સિલેબસ/૧૦૦૪૮/૨૦૨૬
તા.૧૨–૦૫–૨૦૨૬

કુલસચિવ (ઈ.ચા.)

પ્રતિ,
૧) યુનિવર્સિટી સંલગ્ન બી.સી.એ. 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.

# Structure of Program (BCA honors Semester-7)

| Course Category | Course Code | Course Title | Mark sheet Title in English | Level of Course | Teaching Hours/Week | | Exam Duration | | Credit | Internal Marks | | External Marks | | Total Marks |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | TH | PR | TH | PR | | TH | PR | TH | PR | |
| MAJOR | 702-01 | Web Development and design | Web Development and design | 400-499 | 2 | 4 | 1 | 2 | 4 | 25 | 25 | 25 | 25 | 100 |
| | OR | | | | | | | | | | | | | |
| | 702-02 | Android development framework | Android development framework | 400-499 | 2 | 4 | 1 | 2 | 4 | 25 | 25 | 25 | 25 | 100 |
| | 703 | Data Visualization and analytics with tableau | Data Visualization and analytics with tableau | 400-499 | 2 | 4 | 1 | 2 | 4 | 25 | 25 | 25 | 25 | 100 |
| | 704 | Application development using C#.NET | Application development using C#.NET | 400-499 | 2 | 4 | 1 | 2 | 4 | 25 | 25 | 25 | 25 | 100 |
| | 705 | Project | Project | 400-499 | - | 12 | - | 5 | 6 | - | 50 | - | 50 | 100 |
| MINOR | 701 | Version Control and Database Management | Version Control and Database Management | 300-399 | 4 | - | 2 | - | 4 | 50 | - | 50 | - | 100 |

# Structure of Program (BCA honors Semester-8)

| Course Category | Course Code | Course Title | Mark sheet Title in English | Level of Course | Teaching Hours/Week | | Exam Duration | | Credit | Internal Marks | | External Marks | | Total Marks |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | TH | PR | TH | PR | | TH | PR | TH | PR | |
| MAJOR | 802-01 | Web development framework and operations | Web development framework and operations | 400-499 | 2 | 4 | 1 | 2 | 4 | 25 | 25 | 25 | 25 | 100 |
| | OR | | | | | | | | | | | | | |
| | 802-02 | Advanced cross platform app development | Advanced cross platform app development | 400-499 | 2 | 4 | 1 | 2 | 4 | 25 | 25 | 25 | 25 | 100 |
| | 803 | AI implementation using Python | AI implementation using Python | 400-499 | 2 | 4 | 1 | 2 | 4 | 25 | 25 | 25 | 25 | 100 |
| | 804 | Automation Testing framework | Automation Testing framework | 400-499 | 2 | 4 | 1 | 2 | 4 | 25 | 25 | 25 | 25 | 100 |
| | 805 | On the Job Training | On the Job Training | 400-499 | - | 12 | - | 5 | 6 | - | 50 | - | 50 | 100 |
| MINOR | 801 | Open source cloud and database deployment | Open source cloud and database deployment | 300-399 | 4 | - | 2 | - | 4 | 50 | - | 50 | - | 100 |

# Veer Narmad South Gujarat University, Surat

**Computer Science and Information Technology Faculty**

**Syllabus for(Semester-VII and Semester-VIII) of B.C.A. (Honours)**

**As per NEP- 2020**

**To be implemented from Academic Year: June, 2026-2027 (Including Winter Session)**

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

| | |
|---|---|
| **Name of Program:** | Bachelor of Computer Application(Honours) |
| **Abbreviation:** | B.C.A.(Honours): Four-year Integrated Program with Multi-Level Entry and Exit option |
| **Multi-level Exit Criteria:** | i) Under Graduate Certificate in Computer Application: 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.<br>ii) Diploma in Computer Application: 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.<br>iii) B.C.A. (Bachelor's in Computer Application): If the students wish to exit after completion of Third year (Semester-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 B.C.A.(Honors) degree program with multilevel exit options at 1st, 2nd and 3rd Year to obtain Certificate, Diploma, Degree and Honours Degree in Computer Application respectively. |
| **Eligibility:** | Candidate must have passed standard 12th (H.S.C.) Examination in Science (Any Group) / Commerce / vocational / General stream 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.<br><br>In case of candidates passed out from 12th Board from General Stream; Statistics/Economics/Business Mathematics/Accountancy must be one of the subjects. In case of Students passed out with 12th (H.S.C.) vocational stream, Computer and English must be one of the subject. |
| **Objective of the Program:** | Bachelor of Computer Application (BCA) (Honours) is undergraduate degree program in computer application 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.<br><br>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.<br><br>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.<br>**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.<br>**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.<br>**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.<br>**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.<br>**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.<br>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:**

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

**Medium of Instruction:** English

**Program Structure:** Semester-wise Breakup of the course is given as follows:

# Veer Narmad South Gujarat University, Surat
## Program Structure: Fourth Year B.C.A. Honors
## (Semester – 7 and Semester – 8)
**(w.e.f .Academic Year June, 2026-27)**

| **Program Structure** | | Semester-wise breakup for the courses: | | | | |
|---|---|---|---|---|---|---|
| **Semester -7** | | | | | | |
| **Course code** | **Course Title** | **Course Category** | **Level of Course** | **Course Credits** | **Teaching Hours/week** | |
| | | | | **Th+Pra** | **Theory** | **Practical Fieldwork/ Project/ Internship** |
| 701 | Version Control and Database Management (minor-7) | Minor Course | 300-399 | 4 | 4 | 0 |
| 702-01 OR 702-02 | Web Development and design (major-17) OR Android development framework (major-17) | Major Course | 400-499 | 4 | 2 | 4 |
| 703 | Data Visualization and analytics with tableau (major-18) | Major Course | 400-499 | 4 | 2 | 4 |
| 704 | Application development using C#.NET (major-19) | Major Course | 400-499 | 4 | 2 | 4 |
| 705 | Project | Major specific Course | 400-499 | 6 | - | 12 |
| | Practical (Based on Course Code:702-01/702-02, 703 & 704 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 & 704 | | | | |
| **Total** | | | | **22** | **10** | **24** |

| **Course Code** | **Course Title** | **Course Credit** | **University Exam Type** | **Exam Duration** | **External(SEE) Marks** | **Internal (CCE) Marks** | **Total Marks** |
|---|---|---|---|---|---|---|---|
| 701 | Version Control and Database Management | 4 | Theory/ Written | 2 Hours | 50 | 50 | 100 |
| 702-01 OR 702-02 | Web Development and design** OR Android development framework** | 4 | Theory/ Written | 1 Hour | 25 | 25 | 100 |
| | | | Practical | 2 Hours | 25 | 25 | |
| 703** | Data Visualization and analytics with tableau** | 4 | Theory/ Written | 1 Hour | 25 | 25 | 100 |
| | | | Practical | 2 Hours | 25 | 25 | |
| 704** | Application development using C#.NET** | 4 | Theory/ Written | 1 Hour | 25 | 25 | 100 |
| | | | Practical | 2 Hours | 25 | 25 | |
| 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 & 704.**Minimum** Twelve Practical hours (4 hours for course-702-01/702-02, 4 hours for 703 and 4 hours for course-704) 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.

**Major Practical based Subjects: Course 702-01/702-02, 703 & 704 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 (2 hours duration) and course-704 (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 is: Exam evaluation: 20 marks + Viva-voce: 5 Marks.
Practical examination will be conducted for course code-702-01/02,703 and course-704 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 course code–702-01/02,703 and course-704 mandatorily.

| | |
|---|---|
| **Program Passing Rules:** | As per University rules. |

| | |
|---|---|
| Program Fees:<br>(Per Semester)<br>(One time fees and exam fees are additional as prescribed by the university) | Semester Tuition Fees : As per norms of University Semester<br>Laboratory Utilization fees : As per norms of University<br>[Other one time/affiliation/exam fees and other fees under various heads will be as per the norms of the University.]<br>[The fees for certificate courses will be as per the prescribed limit for per credit as per the SOP of certificate courses decided by the University.] |

| Semester-8 | | | | | | |
|---|---|---|---|---|---|---|
| **Course code** | **Course Title** | **Course Category** | **Level of Course** | **Course Credits** | **Teaching Hours/week** | |
| | | | | **Th+Pra** | **Theory** | **Practical Fieldwork/ Project/ Internship** |
| 801 | Open source cloud and database deployment(minor-8) | Minor Course | 300-399 | 4 | 4 | 0 |
| 802-01 OR 802-02 | Web development framework and operations (major-20) OR Advanced cross platform app development (major-20) | Major Course | 400-499 | 4 | 2 | 4 |
| 803 | AI implementation using Python(major-21) | Major Course | 400-499 | 4 | 2 | 4 |
| 804 | Automation Testing framework(major-22) | Major Course | 400-499 | 4 | 2 | 4 |
| 805 | 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 802-01/802-02, 803 & 804 | | | | |
| **Total** | | | | **22** | **10** | **24** |

| **Course Code** | **Course Title** | **Course Credit** | **University Exam Type** | **Exam Duration** | **External Marks** | **Internal Marks** | **Total Marks** |
|---|---|---|---|---|---|---|---|
| 801 | Open source cloud and database deployment | 4 | Theory/ Written | 2 Hours | 50 | 50 | 100 |
| 802-01 OR 802-02 | Web development framework and operations OR Advanced cross platform app development | 4 | Theory/ Written | 1 Hour | 25 | 25 | 100 |
| | | | Practical | 2 Hours | 25 | 25 | |
| 803 | AI implementation using Python | 4 | Theory/ Written | 1 Hour | 25 | 25 | 100 |
| | | | Practical | 2 Hours | 25 | 25 | |
| 804 | Automation Testing framework | 4 | Theory/ Written | 1 Hour | 25 | 25 | 100 |
| | | | Practical | 2 Hours | 25 | 25 | |
| 805 | OJT | 6 | Documentation, Reporting, presentation and viva-voce | Presentation and viva-voce | 50 | 50 | 100 |
| **Total** | | **22** | | | **250** | **250** | **500** |

**For Practical and Project:**
BatchSize–40 Maximum (Desirable). Maximum 45 students can be accommodated in a batch. Separate batch should be considered if the student strength exceed 45 numbers.
Practical includes Practical sessions for course-802-01/802-02, 803 & 804. **Minimum** Twelve Practical hours(4 hours for course-802-01/802-02, 4 hours for 803 and 4 hours for course-804) per week should be allocated per batch.

The journal should be certified by the concerned faculty and by the Head of the Department, failing which the student should 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. The number of courses (subjects) in Major may vary from semester to semester.

**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.

****Major Practical based Subjects:** Course 802-01/802-02, 803 and 804 are major courses. These courses are carrying4 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-802-01/802-02 (2 hours duration), 803(2 hours duration) and course-804(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:07 marks + Home Assignment/Field Assignment:08 marks + Attendance:10 + Class Test*:25

For courses having 25 marks as Internals :
Class Assignment/Active Learning:03 marks + Home Assignment/Field Assignment:03 marks + Attendance:04 + Class Test*:15

**For Practical internal marks(CCE):**
For courses having 25 marks Internals :
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 is: Exam evaluation: 20 marks+Viva-voce:5 Marks.
Practical examination will be conducted for course code-802-01/802-02, 803 and 804 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 course code – 802-01/802-02, 803 and 804 mandatorily.

| | |
|---|---|
| Program Passing Rules: | As per University rules. |
| Program Fees:<br>(Per Semester)<br>(One time fees and exam fees are additional as prescribed by the university) | Semester Tuition Fees : As per norms of University Semester<br>Laboratory Utilization fees : As per norms of University<br>[Other one time/affiliation/exam fees and other fees under various heads will be as per the norms of the University.]<br>[The fees for certificate courses will be as per the prescribed limit for per credit as per the SOP of certificate courses decided by the University.] |

# Course 701: Version Control and Database Management

**[Subject code-2611000907055001]**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** | | | | | | |
| **Semester** | **7th** | | | | | | |
| **NCrF Credit Level** | **6.0** | | | | | | |
| **Course Type** | **Minor** | | | | | | |
| **Course Subtype** | **Nil** | | | | | | |
| **Subject Type** | **Intra-disciplinary** | | | | | | |
| **Course Code** | **701** | | | | | | |
| **Course Level** | **400-499 ( Advanced Level )** | | | | | | |
| **Course Title** | **Version Control and Database Management** | | | | | | |
| **Credit** | **4** | **Theory:** | **4Hrs** | **Practical:** | **-** | **Total:** | **4Hrs** |
| **Effective from:** | **Academic Year:2026–2027** | | | | | | |
| **Course Purpose** | The purpose is to prepare students to work efficiently in software development teams, deploy and manage applications smoothly and handle and analyze data for decision-making. | | | | | | |
| **Course Objectives** | The main objective of this course is:<br>1. Students can understand Version Control Systems (VCS) and their importance in modern software development.<br>2. Students can learn how to use Git and GitHub and gain practical knowledge of branching, merging, and collaborative workflows used in real-world projects.<br>3. Explore containerization using Docker to develop, package, and deploy applications efficiently.<br>4. Students can understand the concepts of Data Warehousing, including ETL processes, data integration, and business intelligence tools. | | | | | | |
| **Pre- requisite** | Knowledge of RDBMS, Python, statistical methods. | | | | | | |
| **Course outcome** | **CO1:** To build a string conceptual understanding of the version control technology, understand necessary functionalities.<br>**CO2:** To learn the concept of Git and its installation and concepts of GitHub. To learn to modify and redistribute the database and keep track of changes using open-source version control systems like Git.<br>**CO3:** To understand the concept of Docker and will learn to package application in containers, allowing them to portable to any system using a Docker container software development platform.<br>**CO4:** To evaluate business needs, design a data warehouse, and integrate and visualize data using dashboards and visual analytics.<br>**CO5:** To learn about Data warehouse process flows and architecture. | | | | | | |

| | PSO1 | PSO2 | PSO3 | PSO4 | PSO5 | PSO6 | PSO7 | PSO8 |
|---|---|---|---|---|---|---|---|---|
| **CO1** | ✓ | ✓ | | ✓ | ✓ | | | |
| **CO2** | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | |
| **CO3** | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| **CO4** | | ✓ | | ✓ | | ✓ | ✓ | ✓ |
| **CO5** | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | |

| Course Content | **Unit-1:Modern Version Control and Git Essentials**<br>1.1. Concepts of Version Control System (VCS)<br>1.1.1. Purpose and importance of Version Control in software development<br>1.1.2. Types of VCS –Centralized vs Distributed<br>1.1.3. Advantages and real-world use cases of VCS in collaborative environments<br>1.2. Introduction to Git<br>1.2.1. Installation and configuration of Git on Windows/Linux/macOS<br>1.2.2. Initializing a Git project and understanding Git directory structure<br>1.2.3. Git file states: Untracked, Modified, Staged, Committed<br>1.3. Git Basic Operations<br>1.3.1. Creating repository, adding files, committing changes<br>1.3.2. Viewing status, logs, and differences<br>1.3.3. Undoing changes and working with. gitignore<br><br>**Unit-2:GitHubandCollaborativeWorkflows**<br>2.1. GitHub Concepts and Repository Hosting<br>2.1.1. Creating GitHub account and repositories<br>2.1.2. Pushing local repository to GitHub<br>2.1.3. Cloning repositories from GitHub<br>2.2. Git Branching and Merging<br>2.2.1. Creating and switching branches<br>2.2.2. Merging branches and resolving conflicts<br>2.2.3. Understanding rebase, stash, and tags<br>2.3. GitHub for Team Collaboration<br>2.3.1. Issues, pull requests, and code reviews<br>2.3.2. Project boards and GitHub Actions (intro to CI/CD)<br>2.3.3. GitHub Desktop and integration with IDEs (VS Code, JetBrains)<br><br>**Unit-3:Containerization Using Docker**<br>3.1. Concepts of Docker<br>3.1.1. Purpose and need for containerization in modern development<br>3.1.2. Installing and setting up Docker Desktop(Windows/macOS/Linux)<br>3.1.3. Docker Architecture and Terminologies<br>3.1.3.1. Images, Containers, Docker Daemon, CLI, DockerHub<br>3.1.3.2. Docker commands: run pull, ps, exec,top<br>3.2. Working with Docker Images and Containers<br>3.2.1. Writing Docker files and building custom images<br>3.2.2. Managing volumes and networks<br>3.2.3. Static site deployment using Docker<br>3.3. Docker Compose and Publishing<br>3.3.1. Introduction to Docker Compose(multi-containerapps)<br>3.3.2. Publishing images to DockerHub<br>3.3.3. Best practices for container management<br><br>**Unit-4:DataWarehousingandIntegration Tools**<br>4.1. Concepts of Data Warehouse<br>4.1.1. Definition, features, and types of data warehouses<br>4.1.2. OLAP vs OLTP –comparative analysis |
|---|---|

| | |
|---|---|
| | 4.1.3. Data Marts, Metadata, Metadata Repository, Data Cube<br>4.2. ETL and Integration Tools<br>4.2.1. Introduction to Extraction, Transformation, Loading (ETL)<br>4.2.2. Data cleaning and transformation approaches<br>4.2.3. Integration of heterogeneous databases<br>4.2.3.1. Query-driven vs Update-driven approaches<br>4.2.3.2. Tools: Overview of Talend/ OpenRefine (demo-based)<br>4.3. Data Warehouse Architecture and Process<br>4.3.1. Data warehouse process flow: Extract, Clean, Load, Backup, Query<br>4.3.2. 3-tier architecture and virtual warehouse concepts<br>4.3.3. Roles of Load Manager, Warehouse Manager, Query Manager<br>4.3.4. Business Analysis Framework and real-world BI use cases |
| **Reference Books** | 1. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd Edition, Ralph Kimball, Margy Ross , ISBN-13: 978- 1118530201, Wiley Inc.<br>2. Database Systems: Introduction to Databases and Data Warehouses 1st Edition, Nenad Jukic, Susan Vrbsky, Svetlozar Nestorov, ISBN-13: 978- 1943153190, Prospect Press<br>3. Building a Scalable Data Warehouse with Data Vault 2.0 - 1st Edition, Daniel Linstedt, Michael Olschimke, ISBN-13: 978-0122025109<br>4. Data Warehousing Fundamentals for IT Professionals 2nd Edition, Paulraj Ponniah, ISBN-13: 978-0410462072, Wiley Inc.<br>5. The Kimball Wiley Inc. Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence Remastered Collection 2nd Edition, ISBN-13: 978-1119216315, Wiley Inc.<br>6. The Pragmatic Programmer: From Journeyman to Master 1st Edition, Andrew Hunt, David Thomas, ISBN-13: 978-0201616224<br>7. Code Complete 2e (Developer Best Practices), Steve Mcconnell, ISBN- 13: 978-0735619678, Microsoft Press US<br>8. The Docker Book, James Turnbull, Publisher: James Turnbull; 18092nd edition<br>9. Docker in Action, 2nd Edition, Jeff Nickoloff, Stephen Kuenzli, ISBN-13: 978-1617294761<br>10. Learning Docker –Second Edition:Build, ship, and scale faster, Jeeva S. Chelladhurai, Vinod Singh, Pethuru Raj, ISBN-13:978-1786462923<br>11. Docker: Up & Running, Karl Matthias, Sean P. Kane, ISBN-13:978- 1491917572 |
| **Teaching Methodology** | Class Work, Discussion, Self-Study, Seminars and/or Assignments |
| **Evaluation Method** | Internal Assessment: 50 Marks.<br>External Assessment: 50 Marks.<br>50% Internal assessment<br>- Attendance, class and home assignment, Unit tests.<br>50% External assessment.<br>- Written Theory Examination |

# Course 702-01 – Web Development and Design

**[**Subject code for Theory-**2611000907011001]** [***Subject code for Practical-2611000907011002***]

<table>
<tr><td>Program Name</td><td colspan="6">Bachelor of Computer Science (Honours)</td></tr>
<tr><td>Semester</td><td colspan="6">7th</td></tr>
<tr><td>NCrF Credit Level</td><td colspan="6">6.0</td></tr>
<tr><td>Course Type</td><td colspan="6">Major</td></tr>
<tr><td>Course Subtype</td><td colspan="6">Nil</td></tr>
<tr><td>Subject Type</td><td colspan="6">Intra-disciplinary</td></tr>
<tr><td>Course Code</td><td colspan="6">702-01</td></tr>
<tr><td>Course Level</td><td colspan="6">400-499 ( Advanced Level )</td></tr>
<tr><td>Course Title</td><td colspan="6">Web Development and Design</td></tr>
<tr><td>Credit</td><td>4</td><td>Theory:</td><td>2 Hrs</td><td>Practical:</td><td>4 Hrs</td><td>Total: 6 Hrs</td></tr>
<tr><td>Effective from:</td><td colspan="6">Academic Year:2026–2027</td></tr>
<tr><td>Course Purpose</td><td colspan="6">The purpose of this course is to provide students with knowledge and practical skills required for modern web application development using JavaScript and React. The course focuses on asynchronous programming, component-based architecture, state management, API integration, routing, and scalable state management using Redux. Students will gain hands-on experience in developing dynamic and interactive web applications using modern front-end technologies.</td></tr>
<tr><td>Course Objectives</td><td colspan="6">1. To introduce advanced JavaScript programming concepts including asynchronous programming and API integration.<br>2. To familiarize students with React framework and component-based architecture for building user interfaces.<br>3. To enable students to manage application state using React Hooks and modern state management techniques.<br>4. To develop skills in handling data, forms, and events in React applications.<br>5. To introduce routing and navigation in React applications.<br>6. To provide knowledge of Redux for managing application-level state in large-scale React applications.</td></tr>
<tr><td>Pre-Requisite</td><td colspan="6">HTML and CSS, JavaScript fundamentals, Basic web development, concepts, Basic understanding of programming logic, Bootstraps fundamentals.</td></tr>
<tr><td>Course Outcomes (COs)</td><td colspan="6">CO1: Apply advanced JavaScript concepts including DOM manipulation, asynchronous programming, and API integration.<br>CO2: Develop interactive user interfaces using React components, JSX, props, and state.<br>CO3: Implement state management and data handling in React applications using React Hooks.<br>CO4: Design scalable web applications using React Router and Redux state management.</td></tr>
<tr><td>CO – PSO Mapping Table</td><td colspan="6"><table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</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>
</table></td></tr>
<tr><td>Content</td><td colspan="6">Unit 1: Java Script and Asynchronous Programming<br>1.1 Working with Objects and Arrays and Functions<br>1.1.1 Object Literals and Methods<br>1.1.2 Array Operations and Iteration Methods</td></tr>
</table>

1.2 User defined functions and arrow functions
1.3 DOM Manipulation
1.3.1 Document Object Model Structure
1.3.2 Event Handling and Event Listeners
1.3.3 Form Handling and Validation
1.4 Advanced JavaScript Concepts
1.4.1 Closures and Scope
1.4.2 Higher Order Functions
1.4.3 Modules and ES6 Features
1.5 Asynchronous JavaScript
1.5.1 Callbacks and Event Loop
1.5.2 Promises and Promise Chaining
1.5.3 Async / Await
1.6 API Integration
1.6.1 Fetch API and Axios
1.6.2 Handling JSON Data
1.6.3 Error Handling in API calls

**Unit 2: React Fundamentals**
2.1. Introduction to React
2.1.1. Virtual DOM Concept
2.1.2. Setting up React Environment
2.2. React Components
2.2.1. Functional Components
2.2.2. JSX Syntax and Rendering
2.3. Props and State
2.3.1. Passing Data via Props
2.4. Integration with Bootstrap

**Unit 3: State Management & Data Handling in React**
3.1. Managing State using useState
3.2. Built-in React Hooks
3.2.1. useEffect, useReducer, useContext
3.2.2. Handling Events and Lifecycle
3.3. Data Handling in React
3.3.1. Fetching Data with useEffect and Fetch API
3.3.2. Handling Loading, Error, and Success States
3.3.3. Working with Forms and Controlled Components
3.4. Rendering Techniques
3.4.1. Conditional Rendering
3.4.2. List Rendering and Keys

**Unit 4: React Navigation and Redux State Management**
4.1. Rout and Navigation
4.1.1. React Router and Navigation
4.1.2. Context API
4.2. Redux Fundamentals
4.2.1. Redux Architecture
4.2.2. Actions, Reducers and Store
4.3. React-Redux Integration
4.3.1. Connecting Components to Store
4.3.2. State Flow in Redux Applications
4.4. Middleware Concepts
4.4.1. Logging Middleware
4.4.2. Async Middleware using Redux Thunk

| | |
|---|---|
| **Teaching Methodology** | Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions. |
| **Evaluation Scheme** | Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks.<br>External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks.<br>50% Internal assessment<br>- Attendance, class and home assignment, Unit tests.<br>- Practical exam, viva-voce, Journal<br>50% External Assessment.<br>- Written theory exam<br>- Practical exam, Viva-Voce |
| **Reference Books** | 1. Modern React with Redux, Stephen Grider, Packt Publishing, 9781786462497, 2017<br>2. Learning React (2nd Edition), Alex Banks; Eve Porcello, O'Reilly Media, 9781492051725, 2020<br>3. React Up and Running, Stoyan Stefanov, O'Reilly Media, 9781491931820, 2016<br>4. Eloquent JavaScript (3rd Edition), Marijn Haverbeke, No Starch Press, 9781593279509, 2018<br>5. You Don't Know JS Yet: Scope and Closures, Kyle Simpson, O'Reilly Media, 9781091210097, 2020<br>6. JavaScript: The Definitive Guide (7th Edition), David Flanagan, O'Reilly Media, 9781491952023, 2020<br>7. Fullstack React: The Complete Guide to ReactJS and Friends, Anthony Accomazzo; Ari Lerner; David Guttman; Nate Murray, Fullstack.io, 9780991344628, 2017<br>8. Beginning React.js: Building Modern Web Applications, Greg Lim, Apress, 9781484232811, 2017<br>9. Programming React: Building Modern Web Applications, Alex Banks; Eve Porcello, O'Reilly Media, 9781492051724, 2020<br>10. Pro React 16, Adam Freeman, Apress, 9781484244500, 2019 |

*[Subject code for Theory-2611000907011003]*

**[Subject code for Practical-2611000907011004]**

# Course 702-02 – Android Development Framework

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** | | | | | | |
| **Semester** | **7th** | | | | | | |
| **NCrF Credit Level** | **6.0** | | | | | | |
| **Course Type** | **Major** | | | | | | |
| **Course Subtype** | **Nil** | | | | | | |
| **Subject Type** | **Intra-disciplinary** | | | | | | |
| **Course Code** | **702-02** | | | | | | |
| **Course Level** | **400-499 ( Advanced Level )** | | | | | | |
| **Course Title** | **Android Development Framework** | | | | | | |
| **Credit** | **4** | **Theory:** | **2 Hrs** | **Practical:** | **4 Hrs** | **Total:** | **6 Hrs** |
| **Effective from:** | **Academic Year:2026–2027** | | | | | | |
| **Course Purpose** | The purpose of this course is to provide students with practical knowledge of Android application development using modern Android frameworks and tools. The course introduces Android architecture, user interface design, data handling, multimedia features, sensors, device connectivity, and deployment of applications. Students will gain hands-on experience in developing feature-rich mobile applications and integrating device services such as location, sensors, and connectivity. | | | | | | |
| **Course Objectives** | 1. To introduce students to Android application development environment and architecture.<br>2. To enable students to design user interfaces and manage application components in Android.<br>3. To develop skills in handling local and remote data using SQLite, JSON, and APIs.<br>4. To provide knowledge of multimedia handling, device connectivity, and sensors in Android applications.<br>5. To familiarize students with Android device services such as location services, maps, and telephony.<br>6. To enable students to test and deploy Android applications to the Google Play Store. | | | | | | |
| **Pre-Requisite** | Java or Kotlin programming, Object-Oriented Programming concepts, Basic understanding of mobile application development | | | | | | |
| **Course Outcomes (COs)** | **CO1:** Develop Android applications using Android Studio, UI components, intents, and fragments.<br>**CO2:** Implement local data storage and API-based data handling in Android applications.<br>**CO3:** Develop multimedia, sensor-based, and device connectivity features in Android applications.<br>**CO4:** Integrate Android device services such as location, maps, and telephony, and deploy applications to the Play Store. | | | | | | |

**Mapping between COs and PSOs**

| | PSO1 | PSO2 | PSO3 | PSO4 | PSO5 | PSO6 | PSO7 | PSO8 |
|---|---|---|---|---|---|---|---|---|
| **CO1** | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | |
| **CO2** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO3** | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO4** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

| | |
|---|---|
| **Content** | **Unit 1: Introduction to Android and Application Components**<br>1.1. Introduction to Mobile Application Development<br>1.2. Android SDK and environment setup<br>1.3. Android Project Structure<br>1.4. Android UI Components<br>1.4.1. Advanced Widgets (TextInputLayout, FloatingActionButton, BottomNavigationView, Snackbar, etc)<br>1.4.2. Layouts in Android (ConstraintLayout, DrawerLayout, CardView)<br>1.5. Android Intents and Intent Filters<br>1.6. Fragments and Fragment Lifecycle<br>1.7. Event handling<br>1.7.1. Input events, Event Listeners<br>1.7.2. Callback Methods (onClick(), onKey(), OnTouch(), OnLongClick(), OnFocusChange(), OnCreateContextMenu(), OnKeyDown(), onTrackballEvent())<br><br>**Unit 2: Data Handling in Android**<br>2.1. Local Data Storage using SQLite<br>2.1.1. Introduction to SQLite in Android<br>2.1.2. CRUD Operations (Insert, Update, Delete, Select)<br>2.2. Shared Preferences<br>2.2.1. Introduction to SharedPreferences<br>2.2.2. Storing and Retrieving Key-Value Data<br>2.2.3. Managing User Sessions<br>2.3. JSON Data Handling<br>2.3.1. Introduction to JSON<br>2.3.2. Parsing JSON Objects and Arrays<br>2.3.3. Converting JSON to Kotlin Objects<br>2.3.4. Introduction to RESTful APIs<br>2.3.5. Sending HTTP Requests (GET, POST)<br>2.3.6. Consuming API Data in Android Applications<br>2.4. Networking Libraries for API Integration<br>2.4.1. Using Retrofit for API calls<br>2.4.2. Using Volley for network requests<br>2.4.3. Handling API Responses<br>2.5. Displaying API Data in UI Components<br>2.5.1. Showing API Data using ListView / RecyclerView<br>2.5.2. Updating UI with Live Data from APIs<br><br>**Unit 3: Multimedia, Sensors and Device Connectivity**<br>3.1. Handling Audio and Video Files<br>3.1.1. Media Player and Media Recorder<br>3.1.2. Methods to handle Audio and Video files (start(), stop(), pause(), setDataSource(), setMediaController(), setVideoURI())<br>3.1.3. Controlling volume and ringer<br>3.1.4. Ringer mode (Normal, Silent, Vibrate)<br>3.2. Text To Speech Class<br>3.2.1. Methods: speak(), setSpeechRate(), setPitch(), setLanguage()<br>3.3. Animation and effect in Android<br>3.3.1. Load and start animation<br>3.3.2. Apply different animations (zoom, Fade, Slide, Rotate)<br>3.4. Handling Camera in Android<br>3.5. Bluetooth API and wifi Connectivity<br>3.5.1. Scan for the available Bluetooth device<br>3.5.2. Connect other device with service discovery<br>3.5.3. Transfer data from one device to another device<br>3.5.4. Scan for wifi network<br>3.5.5. Allow device to connect to the internet |

| | |
|---|---|
| | 3.5.6. Manage list of configured networks<br>3.6. Drag and Drop Framework<br>3.6.1. Drag event and listener<br>3.6.2. Drag and drop processes: start, continuing, dropped, ended<br><br>**Unit 4: Android Device Services and Deployment**<br>4.1. Google Maps API in Android<br>4.2. SMS and Email Integration in Android Apps<br>4.3. Android Telephony Manager<br>4.4. Motion Sensors and Sensor Manager<br>4.4.1. Motion sensor, environmental sensor and position sensor<br>4.4.2. Sensor Framework, classes and interfaces<br>4.4.3. Sensor Manager, Sensor Event<br>4.5. Location Services<br>4.5.1. Fetch and display current location<br>4.5.2. geocoder class and methods to search location on google maps<br>4.6. Android API Usage<br>4.7. Testing Android Applications<br>4.8. Publishing Android Applications to Play Store |
| **Teaching Methodology** | Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions. |
| **Evaluation Scheme** | Internal Assessment: 25 marks Theory + 25 marks Practical = 50 Marks.<br>External Assessment: 25 marks Theory + 25 marks Practical = 50 Marks.<br>50% Internal assessment<br>- Attendance, class and home assignment, Unit tests.<br>- Practical exam, viva-voce, Journal<br>50% External Assessment.<br>- Written theory exam<br>- Practical exam, Viva-Voce |
| **Reference Books** | 1. Android Programming: The Big Nerd Ranch Guide (5th Edition), Bill Phillips; Chris Stewart; Kristin Marsicano, Big Nerd Ranch Guides, 9780135245125, 2019<br>2. Head First Android Development (2nd Edition), Dawn Griffiths; David Griffiths, O'Reilly Media, 9781491974056, 2017<br>3. Android App Development for Dummies, Michael Burton, Wiley Publishing, 9781119017929, 2015<br>4. Professional Android (4th Edition), Reto Meier; Ian Lake, Wiley Publishing, 9781118949528, 2018<br>5. Android Studio Development Essentials – Kotlin Edition, Neil Smyth, Payload Media, 9780997316025, 2022<br>6. Learning Android Application Development, James Talbot; Justin McLean, Packt Publishing, 9781785888366, 2017<br>7. Android Cookbook: Problems and Solutions for Android Developers, Ian F. Darwin, O'Reilly Media, 9781449389697, 2012<br>8. Android Developer Fundamentals Course – Practical Workbook, Google Developers Training Team, Google Developers, 9780998626420, 2017<br>9. Android Studio 4.1 Development Essentials – Kotlin Edition, Neil Smyth, Payload Media, 9781951442231, 2020 |

[Subject code for Theory-2611000907022001]

[Subject code for Practical-2611000907022002]

# Course 703- Data Visualization and analytics with tableau

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** | | | | | | |
| **Semester** | **7th** | | | | | | |
| **NCrF Credit Level** | **6.0** | | | | | | |
| **Course Type** | **Major** | | | | | | |
| **Course Subtype** | **Nil** | | | | | | |
| **Subject Type** | **Intra-disciplinary** | | | | | | |
| **Course Code** | **703** | | | | | | |
| **Course Level** | **400-499 ( Advanced course )** | | | | | | |
| **Course Title** | **Data Visualization and Analytics using Tableau** | | | | | | |
| **Credit** | **4** | **Theory:** | **2 hrs** | **Practical:** | **4 Hrs** | **Total:** | **6 Hrs** |
| **Effective From** | **Academic Year :2026-27** | | | | | | |
| **Course Purpose** | The purpose of this course is to equip students with the knowledge and practical skills required to transform raw data into meaningful visual insights using modern visualization tools. The course focuses on applying visualization principles and analytical techniques through Tableau to support data-driven decision making. Students will learn how to design effective charts, build interactive dashboards, perform data analysis, and communicate insights through visual storytelling. | | | | | | |
| **Course Objectives** | The objectives of this course are to:<br>1. Introduce the fundamental concepts and principles of data visualization.<br>2. Develop practical skills in using Tableau for data exploration and analysis.<br>3. Enable students to create effective and interactive data visualizations.<br>4. Apply analytical techniques such as calculations, forecasting, and clustering within Tableau.<br>5. Develop the ability to design dashboards and communicate insights through visual storytelling. | | | | | | |
| **Pre-requisite** | Students enrolling in this course should have:<br>• Basic knowledge of statistics and data analysis<br>• Familiarity with spreadsheet tools such as Microsoft Excel<br>• Understanding of basic database concepts and data types<br>• Fundamental knowledge of data science concepts | | | | | | |
| **Course Outcomes** | **CO1:** Understand the principles of data visualization and Tableau environment.<br>**CO2:** Create various types of visualizations and apply data transformation techniques.<br>**CO3:** Develop interactive dashboards and apply analytical features for insight generation.<br>**CO4:** Design and implement advanced data visualization solutions using real-world datasets. | | | | | | |

**Mapping between COs and PSOs**

| | PSO1 | PSO2 | PSO3 | PSO4 | PSO5 | PSO6 | PSO7 | PSO8 |
|---|---|---|---|---|---|---|---|---|
| **CO1** | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | |
| **CO2** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO3** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO4** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

| | |
|---|---|
| **Course Content** | **Unit 1: Fundamentals of Data Visualization and Tableau Environment**<br>1.1. Introduction to Data Visualization<br>1.1.1. Importance of Data Visualization<br>1.1.2. Types of Data and Visualization Techniques<br>1.1.3. Principles of Effective Data Visualization<br>1.1.4. Human perception and visual encoding<br>1.2. Introduction to Tableau<br>1.2.1. Tableau ecosystem and components<br>1.2.2. Tableau interface and workspace<br>1.2.3. Tableau Desktop vs Tableau Public<br>1.3. Data Connection and Preparation<br>1.3.1. Connecting to data sources (Excel, CSV, Databases)<br>1.3.2. Understanding Dimensions and Measures<br>1.3.3. Data types and metadata<br>1.3.4. Data cleaning and sorting<br>1.4. Basic Visualizations<br>1.4.1. Bar charts<br>1.4.2. Line charts<br>1.4.3. Pie charts<br>1.4.4. Area charts<br><br>**Unit 2: Core Visualization Techniques and Data Calculations**<br>2.1. Data Manipulation in Tableau<br>2.1.1 Filtering and sorting<br>2.1.2 Groups and sets<br>2.1.3 Continuous vs discrete fields<br>2.2. Advanced Visualization Techniques<br>2.2.1. Scatter plots<br>2.2.2. Histograms<br>2.2.3. Tree maps<br>2.2.4. Heat maps<br>2.2.5. Box plots<br>2.3. Dual Axis and Combined Charts<br>2.3.1. Dual-axis charts<br>2.3.2. Combination charts<br>2.4. Calculated Fields and Table Calculations<br>2.4.1. Creating calculated fields<br>2.4.2. Logical and aggregate functions<br>2.4.3. Running totals<br>2.4.4. Percent difference and ranking<br><br>**Unit 3: Dashboard Design and Advanced Analytics**<br>3.1. Dashboard Development<br>3.1.1. Principles of dashboard design<br>3.1.2. Layout containers and formatting<br>3.1.3. Adding sheets to dashboards<br>3.2. Interactivity in Dashboards<br>3.2.1. Filters and actions<br>3.2.2. Parameters and dynamic analysis<br>3.2.3. Highlighting and drill-down analysis<br>3.3. Analytical Features in Tableau<br>3.3.1. Reference lines and bands<br>3.3.2. Trend lines<br>3.3.3. Forecasting<br>3.3.4. Clustering<br>3.4. Data Integration<br>3.4.1. Data joining<br>3.4.2. Data blending<br>3.4.3. Introduction to Level of Detail (LOD) expressions |

| | |
|---|---|
| | **Unit 4: Data Storytelling, Case Studies and Visualization Projects**<br>4.1. Storytelling with Data<br>4.1.1. Principles of visual storytelling<br>4.1.2. Creating stories in Tableau<br>4.1.3. Designing effective visual narratives<br>4.2. Industry Case Studies<br>4.2.1. Retail analytics dashboards<br>4.2.2. Healthcare data visualization<br>4.2.3. Financial data dashboards<br>4.2.4. Marketing performance dashboards<br>4.3. Visualization Design Best Practices<br>4.3.1. Choosing appropriate visualizations<br>4.3.2. Color theory and visual hierarchy<br>4.3.3. Ethical data visualization<br>4.4. Capstone Visualization Project<br>4.4.1. Dataset selection and preparation<br>4.4.2. Dashboard development<br>4.4.3. Storyboard presentation<br>4.4.4. Project report and interpretation of insights |
| **Reference Books** | 1. Tableau Your Data! Fast and Easy Visual Analysis with Tableau Software, Daniel G. Murray, Wiley publication, 2016, 978-1-119-00119-1.<br>2. Learning Tableau, Joshua N. Milligan, O'Reilly Media, 2019, 978-1-4920-3726-9.<br>3. Tableau Dashboard Cookbook, Andy Kriebel & Ryan Sleeper, Packt Publishing, 2018, 978-1-78839-308-5.<br>4. Storytelling with Data, Cole Nussbaumer Knaflic, Wiley, 2015, 978-1-119-00225-9.<br>5. Communicating Data with Tableau. Ben Jones, O'Reilly Media, 2014, 978-1-4493-7197-2.<br>6. The Big Book of Dashboards, Steve Wexler; Jeffrey Shaffer; Andy Cotgreave, Wiley, 978-1119282716, 2017<br>7. Innovative Tableau, Ryan Sleeper, O'Reilly Media, 978-1491974879, 2018<br>8. Tableau Strategies, Ann Jackson, Packt Publishing, 978-1838824408, 2019<br>9. Practical Tableau, Ryan Sleeper, O'Reilly Media, 978-1491972318, 2018<br>10. Information Dashboard Design, Stephen Few, Analytics Press, 978-1938377006, 2013 |
| **Teaching Methodology** | Class Work, Discussion, Lab work, Self-Study, Seminars and/or Assignments |
| **Evaluation Method** | Internal Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br>External Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br><br>50% Internal assessment.<br>- Attendance, Class and home Assignment, Unit tests.<br>- Practical exam, viva-voce, Journal<br>50% External assessment.<br>- Written Theory exam<br>- Practical Exam, viva-voce |

**[Subject code for Theory-2611000907033001]**

**[Subject code for Practical-2611000907033002]**

# Course 704- Application Development using C#.NET

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** | | | | | | |
| **Semester** | **7th** | | | | | | |
| **NCrF Credit Level** | **6.0** | | | | | | |
| **Course Type** | **Major** | | | | | | |
| **Course Subtype** | **Nil** | | | | | | |
| **Subject Type** | **Intra-disciplinary** | | | | | | |
| **Course Code** | **704** | | | | | | |
| **Course Level** | **400-499 ( Advanced Level )** | | | | | | |
| **Course Title** | **Application Development using C#.NET** | | | | | | |
| **Credit** | **4** | **Theory:** | **2 hrs** | **Practical:** | **4 Hrs** | **Total:** | **6 Hrs** |
| **Effective From** | **Academic Year :2026-27** | | | | | | |
| **Course purpose** | The purpose of this course is to equip students with the knowledge and practical skills required to design and develop modern software applications using the C# programming language and the .NET development platform. The course introduces students to the fundamentals of C# programming, object-oriented programming concepts, and the architecture of the .NET ecosystem.<br>It further enables students to build scalable and maintainable applications using ASP.NET Core and the Model–View–Controller (MVC) framework. Students will also gain experience in developing data-driven applications using Entity Framework Core and creating RESTful Web APIs.<br>By the end of the course, learners will be capable of designing, implementing, and deploying enterprise-level applications while following industry-standard development practices, thereby preparing them for careers in software development and web application development. | | | | | | |
| **Course Objectives** | 1. To introduce students to C# programming and the .NET platform.<br>2. To develop object-oriented programming skills using C#.<br>3. To build web applications using ASP.NET Core MVC.<br>4. To implement modern .NET technologies including Web APIs, EF Core, and authentication. | | | | | | |
| **Pre-requisite** | Students enrolling in this course are expected to have basic knowledge of the following:<br>1. Fundamentals of Programming using languages such as C, C++, Java, or Python.<br>2. Basic understanding of Object-Oriented Programming concepts such as classes, objects, inheritance, and polymorphism.<br>3. Basic knowledge of data structures and algorithms.<br>4. Basic understanding of database concepts and SQL.<br>5. Familiarity with web technologies such as HTML and basic web architecture is desirable. | | | | | | |
| **Course Outcomes** | After completion of this course, students will be able to:<br>**CO1:** Explain the architecture of .NET and write basic C# programs.<br>**CO2:** Apply object-oriented programming concepts using C#.<br>**CO3:** Develop web applications using ASP.NET Core MVC.<br>**CO4:** Build data-driven applications using Entity Framework and Web APIs. | | | | | | |

**Mapping between COs and PSOs**

| | PSO1 | PSO2 | PSO3 | PSO4 | PSO5 | PSO6 | PSO7 | PSO8 |
|---|---|---|---|---|---|---|---|---|
| **CO1** | ✓ | ✓ | | ✓ | ✓ | | ✓ | |
| **CO2** | ✓ | ✓ | | ✓ | ✓ | | ✓ | |
| **CO3** | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| **CO4** | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**Course Content**

**Unit 1: Fundamentals of C# and .NET Platform**

1.1. Introduction to .NET Framework and .NET Core
- 1.1.1. Overview of .NET Architecture
- 1.1.2. Assemblies and .NET Compilation Process

1.2. Basics of C# Programming
- 1.2.1. Structure of C# Program
- 1.2.2. Variables, Data Types, and Operators
- 1.2.3. Control Statements (if, switch, loops)
- 1.2.4. Arrays and Strings

1.3. Object-Oriented Programming in C#
- 1.3.1. Classes and Objects
- 1.3.2. Constructors and Destructors
- 1.3.3. Encapsulation and Abstraction
- 1.3.4. Inheritance and Polymorphism

**Unit 2: Advanced C# Programming**

2.1. Exception Handling
- 2.1.1. try–catch–finally blocks
- 2.1.2. Custom Exceptions

2.2. Collections and Generics
- 2.2.1. List, Dictionary, Queue, Stack
- 2.2.2. Generic Classes and Methods

2.3. Delegates and Events
- 2.3.1. Delegates and Multicast Delegates
- 2.3.2. Events and Event Handling

2.4. Language Integrated Query (LINQ)
- 2.4.1. LINQ Basics
- 2.4.2. LINQ to Collections

**Unit 3: ASP.NET Core and MVC Architecture**

3.1. Introduction to ASP.NET Core
- 3.1.1. ASP.NET Core Architecture
- 3.1.2. Project Structure

3.2. Model-View-Controller (MVC) Framework
- 3.2.1. MVC Architecture
- 3.2.2. Controllers and Actions
- 3.2.3. Routing

3.3. Views and Razor Syntax
- 3.3.1. Razor View Engine
- 3.3.2. Layouts and Partial Views

3.4. Forms and Model Binding
- 3.4.1. Form Handling
- 3.4.2. Model Validation

**Unit 4: Advanced .NET Application Development**

4.1. Entity Framework Core
- 4.1.1. ORM Concepts
- 4.1.2. Code First Approach
- 4.1.3. CRUD Operations using EF Core

| | |
|---|---|
| | 4.2. Web API Development<br>4.2.1. REST Architecture<br>4.2.2. Creating Web APIs in ASP.NET Core<br>4.2.3. API Testing using Swagger/Postman |
| **Reference Books** | 1. C# 10 and .NET 6 – Modern Cross-Platform Development, Mark J. Price, Packt Publishing, 2021, ISBN-13: 9781801077361<br>2. Pro ASP.NET Core MVC, Adam Freeman, Apress, 2016, ISBN-13: 9781484205303.<br>3. ASP.NET Core in Action, Andrew Lock, Manning Publications, 2018, ISBN-13: 9781617294617.<br>4. Programming C#, Ian Griffiths, O'Reilly Media, 2022, ISBN-13: 9781098117818.<br>5. Head First C#, Andrew Stellman & Jennifer Greene, O'Reilly Media, 2021, ISBN-13: 9781491976708<br>6. Programming in C# with .NET, E. Balagurusamy, McGraw Hill Education India, 2017, 9789352602452<br>7. ASP.NET with C#, S. Thamarai Selvi and R. Murugesan, Pearson Education India, 2018, 9789332585461<br>8. .NET Programming using C#, Kogent Learning Solutions Inc., Dreamtech Press, 2013, 9789351193364<br>9. C# and .NET Framework, B.M. Harwani, BPB Publications, 2019, 9789388511643<br>10. Mastering ASP.NET with C#, B.M. Harwani, BPB Publications, 2017, 9789387284494. |
| **Teaching Methodology** | Class Work, Discussion, Lab work, Self-Study, Seminars and/or Assignments |
| **Evaluation Method** | Internal Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br>External Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br><br>50% Internal assessment.<br>- Attendance, Class and home Assignment, Unit tests.<br>- Practical exam, viva-voce, Journal<br>50% External assessment.<br>- Written Theory exam<br>- Practical Exam, viva-voce |

[Subject code-2611000907044001]

# Course 705: Project

| | |
|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** |
| **Semester** | **7th** |
| **NCrF Credit Level** | **6.0** |
| **Course Type** | **Major** |
| **Course Subtype** | **Nil** |
| **Subject Type** | **Intra-disciplinary** |
| **Course Code** | **705** |
| **Course Level** | **400-499 ( Advanced Level )** |
| **Course Title** | **Project** |
| **Credit** | **6** |
| **Effective From** | **Academic Year :2026-27** |
| **Course Purpose** | This course is designed to provide students with the opportunity to apply the knowledge and skills they have gained throughout their academic journey in web design, mobile applications, and web technologies. It encourages hands-on learning by developing a real world, full-scale project through self-exploration of technologies, structured documentation, and effective presentation. Project is based on Course 702-01 and 704 in case the student has opted web development electives. In option to this, the students who selected elective as android based course, the project will be based on 702-02 and 704. |
| **Course Objective** | (1). Understand and analyze the given project definition and plan development accordingly.<br>(2). Apply learned and self-acquired knowledge of technologies in designing and implementing project solutions.<br>(3). Demonstrate the use of appropriate tools, frameworks, and platforms in project development.<br>(4). Develop a well-structured project document covering all phases of the development life cycle.<br>(5). Present the project effectively using professional communication and presentation tools. |
| **Pre-requisite** | Students must have completed foundational and intermediate courses in web design, mobile application development, and web technologies. They should be familiar with programming languages (such as HTML, CSS, JavaScript, Web-technologies/Mobile Technologies, Python, or Java), database concepts, and basic software development practices. Prior experience with mini-projects or assignments involving real-world problem-solving is desirable. |
| **Course Outcomes** | **CO1:** Analyze: Students will be able to analyze project requirements, identify suitable tools, and prepare an implementation strategy.<br>**CO2:** Create: Students will develop full-fledged applications using relevant web, mobile, or hybrid technologies.<br>**CO3:** Apply: Students will gain experience in applying the Software Development Life Cycle (SDLC) to real-world problems.<br>**CO4:** Create: Students will prepare and submit a comprehensive project report that meets academic and professional standards.<br>**CO5:** Evaluate: Students will present their project solutions confidently and clearly to technical and non-technical audiences. |

<table>
<tr><td>Project Development:</td><td>
STEP-1: Project Planning and Definition<br>
1.1. Understanding Problem Statement<br>
1.2. Feasibility Study and Requirement Analysis<br>
1.3. Technology Stack Selection (Web, Mobile, Cloud, Database)<br>
1.4. Project Scheduling and Team Role Allocation<br><br>
STEP-2: Project Design and Architecture<br>
2.1. System Design – High Level and Low Level<br>
2.2. Database Design and ER Diagram<br>
2.3. UI/UX Planning and Wireframing<br>
2.4. Data Flow Diagram and Architecture Diagram<br><br>
STEP-3: Project Development<br>
3.1. Frontend Development<br>
3.2. Backend Development<br>
3.3. Integration with Database and External APIs<br>
3.4. Testing: Unit Testing, Integration Testing, User Acceptance Testing<br><br>
STEP-4: Documentation and Deployment<br>
4.1. Preparing Project Documentation: SRS, Design Document, User Manual<br>
4.2. Deployment on Hosting Platforms (like Firebase, Heroku, GitHub Pages, etc.)<br>
4.3. Project Report Writing in Standard Format<br>
4.4. Preparing and Delivering Project Presentation<br><br>
[Students will submit E-Document/Hardcopy for Project report. One internal guide will be allocated for every ten groups All groups are required to contact their internal guides once a week to endorse their project progress work.]
</td></tr>
<tr><td>Project Evaluation Scheme:</td><td>
<table>
<tr><th>Component</th><th>Marks</th></tr>
<tr><td>Problem Definition and Planning</td><td>10%</td></tr>
<tr><td>Design and Architecture</td><td>15%</td></tr>
<tr><td>Implementation and Functionality</td><td>30%</td></tr>
<tr><td>Testing and Deployment</td><td>15%</td></tr>
<tr><td>Documentation</td><td>10%</td></tr>
<tr><td>Final Presentation & Viva</td><td>20%</td></tr>
<tr><td>Total</td><td>100%</td></tr>
</table>
</td></tr>
<tr><td>Teaching Methodology</td><td>Applied knowledge, External project work, Lab. Work, hands-on-experience, webinar, seminar, demonstrations, expert lectures.</td></tr>
<tr><td>Evaluation Method</td><td>50% Internal assessment.<br>- Attendance and reporting to internal guides<br>- Internal project presentation and demonstration, project documentation.<br>50% External assessment.<br>- Project presentation and demonstration, viva-voce and e-project/hardcopy report.</td></tr>
</table>

# Course 801: Open Source Cloud and Database Deployment

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** | | | | | | |
| **Semester** | **8th** | | | | | | |
| **NCrF Credit Level** | **6.0** | | | | | | |
| **Course Type** | **Minor** | | | | | | |
| **Course Subtype** | **Nil** | | | | | | |
| **Subject Type** | **Intra-disciplinary** | | | | | | |
| **Course Code** | **801** | | | | | | |
| **Course Level** | **400-499 ( Advanced Level )** | | | | | | |
| **Course Title** | **Open Source Cloud and Database Deployment** | | | | | | |
| **Credit** | **4** | **Theory:** | **4 hrs** | **Practical:** | **-** | **Total:** | **4 Hrs** |
| **Effective From** | **Academic Year :2026-27** | | | | | | |

| | |
|---|---|
| **Course Purpose** | The course "Open Source Cloud and Database Deployment" is designed to equip students with foundational and practical knowledge of deploying applications and databases using open-source cloud platforms. It introduces learners to essential concepts of cloud computing, helping them understand how to work with virtual machines, manage data remotely, and integrate cloud-hosted databases into real-world applications. The course emphasizes hands-on experience in uploading, organizing, and maintaining datasets on the cloud while using tools that require no proprietary licenses. Students will learn to connect applications with cloud-based databases and ensure smooth data flow between client and server components. A key focus is placed on deploying both static and dynamic websites and full-stack projects using open-source infrastructure. This course empowers students to independently manage cloud-based projects from development to deployment, making them industry-ready for cloud-centric roles. |
| **Course Objectives** | The main objective of this course is to:<br>1. Students can understand the fundamentals of cloud computing and open-source cloud platforms.<br>2. Students can learn how to set up and manage virtual machines and cloud storage systems**.**<br>3. Students can gain practical knowledge of uploading, managing, and querying data on cloud databases**.**<br>4. To develop skills for integrating backend and frontend applications with cloud databases**.**<br>5. To learn techniques for deploying static and full-stack applications on cloud platforms**.**<br>6. To understand docker-based deployment, CI/CD basics, and post-deployment management. |
| **Course Outcome** | **CO1:** Understand and explain the fundamentals of cloud computing and open source cloud platforms.<br>**CO2:** Demonstrate the ability to upload and manage datasets on open cloud infrastructure.<br>**CO3:** Apply techniques to maintain and manipulate cloud hosted databases and tables.<br>**CO4:** Integrate cloud-based datasets with full-stack applications using open source tools.<br>**CO5:** Deploy static and dynamic websites or projects on open source cloud environments. |

| | |
|---|---|
| **Pre-requisite** | Students should have basic knowledge of web development using HTML, CSS, JavaScript, and fundamentals of back end development using Node.js.Familiarity with Git, GitHub, and basic database concepts is recommended for Effective understanding and hands-on practice. |

**Mapping between COs and PSOs**

| | PSO1 | PSO2 | PSO3 | PSO4 | PSO5 | PSO6 | PSO7 | PSO8 |
|---|---|---|---|---|---|---|---|---|
| **CO1** | ✓ | ✓ | | ✓ | ✓ | | ✓ | |
| **CO2** | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| **CO3** | | ✓ | | ✓ | ✓ | ✓ | ✓ | |
| **CO4** | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO5** | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**Course Content**

**Unit-1: Introduction to Open Source Cloud**

1.1. Fundamentals of Cloud Computing
- 1.1.1. Definition and Characteristics of Cloud Computing
- 1.1.2. Cloud Service Models: IaaS, PaaS, SaaS
- 1.1.3. Deployment Models: Public, Private, Hybrid, Community Cloud
- 1.1.4. Advantages of using Open Source Cloud Solutions

1.2. Open Source Cloud Platforms
- 1.2.1. Introduction to platforms like Linode, DigitalOcean, and OpenStack
- 1.2.2. Setting up a Virtual Machine (VM)on open cloud platforms
- 1.2.3. SSH access, security setup, and remote server basics

1.3. Cloud Storage Basics
- 1.3.1. Concepts of cloud-based file systems and object storage
- 1.3.2. Working with open-source storage tools(e.g.,NextCloud, MinIO)

**Unit-2: Uploading and Managing Data on Cloud**

2.1. Data Preparation and Format
- 2.1.1. CSV, JSON, and SQL dump formats
- 2.1.2. Structuring data for cloud compatibility

2.2. Uploading Datasets to Cloud
- 2.2.1. Using SCP, SFTP, and cloud dashboards
- 2.2.2. Uploading tables to cloud databases (e.g.,PostgreSQL, MariaDB)
- 2.2.3. Importing datasets using CLI and GUI(e.g.,Adminer, DBeaver)

2.3. Managing Cloud-hosted Databases
- 2.3.1. Creating and modifying tables
- 2.3.2. Running SQL queries remotely
- 2.3.3. Backup and recovery of datasets

**Unit-3: Application and Dataset Integration**

3.1. Backend Integration
- 3.1.1. Connecting cloud databases with Node.js/Express applications
- 3.1.2. Environment variables and security practices(dotenv)
- 3.1.3. Using ORMs (Sequelize/Mongoose)for database operations

3.2. Frontend Data Display
- 3.2.1. Fetching data from backend APIs
- 3.2.2. Displaying dynamic data using React components
- 3.2.3. Error handling and live refresh

3.3. Real-time Database Interaction
- 3.3.1. API calls using Axios/Fetch
- 3.3.2. Form submission and storage
- 3.3.3. Validation and data binding

| | |
|---|---|
| | **Unit-4: Cloud-based Deployment of Projects**<br>4.1. Deployment of Static Websites<br>4.1.1. Hosting using GitHub Pages or Netlify(for frontend-only apps)<br>4.1.2. CI/CD overview and automation for static deployment<br>4.2. Deployment of Full Stack Projects<br>4.2.1. Using open-source PaaS: Render, Railway, CapRover<br>4.2.2. Deploying Node.js apps with database connection<br>4.2.3. Dockerizing full stack applications for cloud deployment<br>4.3. Post-deployment Operations<br>4.3.1. Monitoring, log management using open tools(e.g., Logtail, Grafana basics)<br>4.3.2. Common deployment issues and trouble shooting<br>4.3.3. Maintenance and update of live websites |
| **Reference Books** | 1. Cloud Computing: Concepts, Technology & Architecture by Thomas Erl, published by Pearson Education, ISBN: 9780133387520.<br>2. Learning MySQL and MariaDB: Heading in the Right Direction with MySQL and MariaDB by Russell J.T. Dyer, published by O'Reilly Media, ISBN: 9781449357283.<br>3. Docker: Up &Running by Karl Matthias and Sean P. Kane, published by O'Reilly Media, ISBN: 9781492036739.<br>4. Node.js Web Development by David Herron, published by Packt Publishing, ISBN: 9781838987575.<br>5. Learning React: Functional Web Development with React and Redux by Alex Banks and Eve Porcello, published by O'Reilly Media, ISBN: 9781492051725.<br>6. Database Systems: Models, Languages, Design and Application Programming by Ramez Elmasri and Shamkant B. Navathe, published by Pearson Education, ISBN: 9789332582705.<br>7. Fundamentals of Database Systems by C. J. Date, published by Pearson Education, ISBN: 9780133970777.<br>8. Cloud Computing by Dr. Kumar Saurabh, published by Wiley India Pvt. Ltd., ISBN: 9788126547295.<br>9. Cloud Computing: A Hands-on Approach by Arshdeep Bahga and Vijay Madisetti, published by University Press, ISBN: 9780996025515.<br>10. OpenStack in Action by Cody Bumgardner, published by Manning Publications, ISBN: 9781617292163. |
| **Teaching Methodology** | Class Work, Discussion, Self-Study, Seminars and/or Assignments |
| **Evaluation Method** | Internal Assessment :50 Marks Theory<br>External Assessment :50 Marks Theory<br><br>50%Internal assessment.<br>- Internal Unit tests, class and home assignments and attendance<br><br>50% External assessment.<br>- Written Theory Examination |

**[Subject code for Theory-2711000908011001]**

**[Subject code for Practical-2711000908011002]**

# Course 802-01 – Web Development Framework and Operations

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** | | | | | | |
| **Semester** | **8th** | | | | | | |
| **NCrF Credit Level** | **6.0** | | | | | | |
| **Course Type** | **Major** | | | | | | |
| **Course Subtype** | **Nil** | | | | | | |
| **Subject Type** | **Intra-disciplinary** | | | | | | |
| **Course Code** | **802-01** | | | | | | |
| **Course Level** | **400-499 ( Advanced Level )** | | | | | | |
| **Course Title** | **Web Development Framework and Operations** | | | | | | |
| **Credit** | **4** | **Theory:** | **2 hrs** | **Practical:** | **4 hrs** | **Total:** | **6 Hrs** |
| **Effective From** | **Academic Year :2026-27** | | | | | | |

**Course Purpose**

The purpose of this course is to provide students with practical knowledge of modern full-stack web development and DevOps practices. The course introduces front-end development using Angular, backend development using Node.js and Express.js, and modern software development methodologies including Agile and DevOps. Students will also learn configuration management, containerization, and CI/CD pipelines to automate the deployment and monitoring of web applications.

**Course Objectives**

1. To introduce the Angular framework for building dynamic and scalable front-end web applications.
2. To develop backend applications and REST APIs using Node.js and Express.js.
3. To familiarize students with Agile development methodologies used in modern software projects.
4. To introduce DevOps concepts including CI/CD, containerization, and cloud deployment.
5. To enable students to implement automated deployment pipelines and monitoring systems.

**Pre-Requisite**

HTML and CSS, JavaScript fundamentals, Basic web development, concepts, Basic understanding of programming logic

**Course Outcomes (COs)**

**CO1:** Develop dynamic web applications using Angular components, directives, and forms.

**CO2:** Build backend services and RESTful APIs using Node.js and Express.js.

**CO3:** Apply Agile methodologies and DevOps principles in software development workflows.

**CO4:** Implement CI/CD pipelines, containerization, and monitoring tools for automated application deployment and management.

**CO – PSO Mapping Table**

| | PSO1 | PSO2 | PSO3 | PSO4 | PSO5 | PSO6 | PSO7 | PSO8 |
|---|---|---|---|---|---|---|---|---|
| **CO1** | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO2** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO3** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO4** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**Content**

**Unit 1: Fundamentals of Angular Framework**

1.1. Introduction to Angular

1.1.1. Angular Architecture

1.1.2. Angular CLI Setup

| | |
|---|---|
| | 1.2. Angular Components and Modules<br>1.2.1. Component Lifecycle<br>1.2.2. Data Binding Techniques (one-way, two-way)<br>1.3. Directives and Pipes<br>1.3.1. Structural and Attribute Directives (*ngIf, *ngFor, ngClass, ngStyle)<br>1.3.2. Built-in and Custom Pipes<br>1.3.3. User Input and Events (click, mouse, key events)<br>1.4. Forms and Validation<br>1.4.1. Template Driven Forms<br>1.4.2. Reactive Forms<br>1.5. AJAX and JSON Data handling in Angular<br><br>**Unit 2: Backend Development using Express.js and Node.js**<br>2.1. Node.js Fundamentals<br>2.1.1. Event-driven Architecture<br>2.1.2. NPM and Package Management<br>2.2. Express.js Framework<br>2.2.1. Setting up Express Applications<br>2.2.2. Routing and Middleware<br>2.3. REST API Development<br>2.3.1. CRUD Operations<br>2.3.2. JSON Data Handling<br>2.4. Authentication and Session Handling<br>2.4.1. Cookies and Sessions<br>2.4.2. Basic Security Practices<br><br>**Unit 3: Agile Methodologies & DevOps Fundamentals**<br>3.1. Agile Methodologies in Web Development<br>3.1.1. Scrum and Sprint Planning<br>3.1.2. Kanban and Agile Workflows<br>3.1.3. User Stories and Backlog Management<br>3.1.4. Agile Testing and Continuous Feedback<br>3.2. DevOps Concepts<br>3.2.1. DevOps Lifecycle and Culture<br>3.2.2. Continuous Integration and Continuous Deployment<br>3.2.3. Shift-Left Testing and DevSecOps Basics<br>3.2.4. Monitoring and Logging in DevOps<br>3.3. DevOps Tools Overview<br>3.3.1. Git and Version Control<br>3.3.2. Maven and Build Automation<br>3.3.3. Docker Basics and Containerization<br>3.3.4. Introduction to Cloud Platforms (AWS / Azure)<br><br>**Unit 4: Configuration Management & CI/CD Pipelines**<br>4.1. Configuration Management<br>4.1.1. Introduction to Ansible<br>4.1.2. Writing Ansible Playbooks<br>4.1.3. Managing Inventory and Roles in Ansible<br>4.1.4. Introduction to Terraform for Infrastructure as Code<br>4.2. CI/CD using Jenkins<br>4.2.1. Jenkins Architecture<br>4.2.2. Creating Pipelines for Web Applications<br>4.2.3. Integrating Jenkins with Git and Maven<br>4.2.4. Automated Testing in Jenkins Pipelines |

| | |
|---|---|
| | 4.3. Containerization & Orchestration<br>4.3.1. Docker Images, Containers and Volumes<br>4.3.2. Docker Compose for Multi-Container Apps<br>4.3.3. Introduction to Kubernetes<br>4.3.4. Deploying Applications with Kubernetes<br>4.4. Monitoring & DevOps Security<br>4.4.1. Application Monitoring with Prometheus &Grafana<br>4.4.2. Log Management using ELK Stack<br>4.4.3. DevSecOps: Security in CI/CD Pipelines<br>4.4.4. Incident Management and Alerting |
| **Teaching Methodology** | Classroom Teaching, Hands-on Lab Sessions, Guided Project Work, Demonstrations, Assignments, and Discussions. |
| **Evaluation Scheme** | Internal Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br>External Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br><br>50% Internal Assessment:<br>- Attendance, class and home Assignments, Unit Tests<br>- Practical exam, viva-voce, Journal<br>50% External Assessment:<br>- Written Theory Examination<br>- Practical exam, Viva-Voce |
| **Reference Books** | 1. Learning Angular (4th Edition), Aristeidis Bampakos; Pablo Deeleman, Packt Publishing, 9781803233444, 2023<br>2. Angular: Up and Running, Shyam Seshadri, O'Reilly Media, 9781491999837, 2018<br>3. Node.js Web Development (5th Edition), David Herron, Packt Publishing, 9781803235240, 2023<br>4. Web Development with Node and Express (2nd Edition), Ethan Brown, O'Reilly Media, 9781492053514, 2019<br>5. Agile Software Development with Scrum, Ken Schwaber; Mike Beedle, Prentice Hall, 9780130676344, 2001<br>6. The DevOps Handbook, Gene Kim; Jez Humble; Patrick Debois; John Willis, IT Revolution Press, 9781942788003, 2016<br>7. Docker Deep Dive, Nigel Poulton, Lean Publishing, 9781521822807, 2020<br>8. Jenkins: The Definitive Guide, John Ferguson Smart, O'Reilly Media, 9781449303358, 2011<br>9. Pro Angular (2nd Edition), Adam Freeman, Apress, 9781484241790, 2019<br>10. Kubernetes Up & Running: Dive into the Future of Infrastructure (3rd Edition), Brendan Burns; Joe Beda; Kelsey Hightower, O'Reilly Media, 9781098110208, 2022 |

**[Subject code for Theory-2711000908011003]**

**[Subject code for Practical-2711000908011004]**

# Course 802-02 – Advanced Cross Platform App Development

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** | | | | | | |
| **Semester** | **8th** | | | | | | |
| **NCrF Credit Level** | **6.0** | | | | | | |
| **Course Type** | **Major** | | | | | | |
| **Course Subtype** | **Nil** | | | | | | |
| **Subject Type** | **Intra-disciplinary** | | | | | | |
| **Course Code** | **802-02** | | | | | | |
| **Course Level** | **400-499 ( Advanced Level )** | | | | | | |
| **Course Title** | **Advanced Cross Platform App Development** | | | | | | |
| **Credit** | **4** | **Theory:** | **2 hrs** | **Practical:** | **4 hrs** | **Total:** | **6 Hrs** |
| **Effective From** | **Academic Year :2026-27** | | | | | | |

**Course Purpose**

The purpose of this course is to provide students with knowledge and practical skills in cross-platform mobile application development using modern frameworks such as Flutter and React Native. The course focuses on UI design, navigation, API integration, local storage, state management, and application deployment. Students will gain hands-on experience in building high-performance mobile applications that run on multiple platforms using a single codebase.

**Course Objectives**

1. To introduce cross-platform mobile application development concepts and tools.
2. To enable students to develop mobile applications using Flutter and Dart programming.
3. To familiarize students with navigation, API integration, and local data storage in Flutter applications.
4. To introduce React Native architecture and component-based mobile UI development.
5. To develop skills in building and deploying cross-platform applications using modern frameworks.

**Pre-Requisite**

Programming concepts (JavaScript / Dart / Java or Kotlin basics, Object-Oriented Programming concepts, Basic understanding of mobile application development

**Course Outcomes (COs)**

**CO1:** Develop cross-platform mobile user interfaces using Flutter widgets and layout techniques.

**CO2:** Implement navigation, API integration, local data storage, and animations in Flutter applications.

**CO3:** Develop mobile applications using React Native components, state management, and styling techniques.

**CO4:** Build, test, and deploy cross-platform mobile applications using Flutter and React Native frameworks.

**CO – PSO Mapping Table**

| | PSO1 | PSO2 | PSO3 | PSO4 | PSO5 | PSO6 | PSO7 | PSO8 |
|---|---|---|---|---|---|---|---|---|
| **CO1** | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO2** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO3** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO4** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**Content**

**Unit 1: Introduction to Cross Platform Development and Flutter**

1.1. Introduction to Cross Platform Mobile Development
1.2. Overview of Flutter Framework
1.3. Flutter Architecture

1.4. Setting up Flutter Development Environment
1.5. Flutter Widgets
    1.5.1. Stateless Widgets (Text, Icon, Image, Container, Row, Column, Center, Padding, SizedBox, Card, Scaffold, AppBar, Divider, Spacer, RichText)
    1.5.2. Stateful Widgets (TextField, Checkbox, Radio, Switch, Slider, Form, ListView, GridView, PageView, TabBar, TabBarView, Drawer, BottomNavigationBar, AlertDialog, SnackBar)
1.6. Layout and UI Design in Flutter
    1.6.1. basic layout widgets (Container, SizedBox, Padding, Center)
    1.6.2. Row and Column Layout
    1.6.3. Flexible Layout (Expanded, Flexible, Spacer)
    1.6.4. Scaffold, AppBar, Card, Stack
    1.6.5. Responsive UI Design (MediaQuery, LayoutBuilder)

**Unit 2: Flutter Advanced Concepts**
2.1. Navigation and Routing in Flutter
    2.1.1. Introduction to Navigation
    2.1.2. Navigator and Routes
    2.1.3. Passing Data between Screens
2.2. REST API Integration in Flutter
    2.2.1. Introduction to RESTful APIs
    2.2.2. Fetching Data from Web APIs
    2.2.3. Handling API Responses
2.3. Handling HTTP Requests
    2.3.1. GET Request
    2.3.2. POST Request
    2.3.3. Error Handling in HTTP requests
2.4. JSON Data Handling
    2.4.1. JSON Structure and Format
    2.4.2. Parsing JSON Data in Flutter
    2.4.3. Converting JSON to Dart Objects
2.5. Local Data Storage in Flutter (SQLite)
    2.5.1. Introduction to SQLite
    2.5.2. Creating and Managing Database
    2.5.3. Executing SQL Statements (Insert, Update, Delete, Select)
2.6. Flutter Animation
    2.6.1. Introduction to Animation in Flutter
    2.6.2. Implicit Animations
    2.6.3. Explicit Animations

**Unit 3: React Native Fundamentals**
3.1. Introduction to React Native
    3.1.1. Overview of Cross-Platform Mobile Development
    3.1.2. Features and Advantages of React Native
    3.1.3. Setting up React Native Development Environment
3.2. React Native Architecture
    3.2.1. Architecture Overview
    3.2.2. Bridge Concept
    3.2.3. Interaction between JavaScript and Native Modules
3.3. Components in React Native
    3.3.1. Functional Components
    3.3.2. Class Components
    3.3.3. Component Lifecycle Basics
3.4. State and Props
    3.4.1. Understanding Props

| | |
|---|---|
| | 3.4.2. Managing State in Components<br>3.4.3. Updating State and Re-rendering<br>3.5. React Native UI Components: View, Text, Image, ScrollView, Button, TextInput<br>3.6. Styling and Layout in React Native<br>3.6.1. Styling Components using StyleSheet<br>3.6.2. Flexbox Layout System<br>3.6.3. Responsive UI Design<br><br>**Unit 4: Advanced React Native Application Development**<br>4.1. Navigation in React Native<br>4.1.1. Introduction to Navigation<br>4.1.2. Stack Navigation<br>4.1.3. Tab Navigation<br>4.1.4. Passing Data Between Screens<br>4.2. Handling Forms and User Input<br>4.2.1. TextInput Handling<br>4.2.2. Form Validation<br>4.2.3. Handling User Events<br>4.3. Working with APIs in React Native<br>4.3.1. Introduction to REST APIs<br>4.3.2. Fetching Data using Fetch API or Axios<br>4.3.3. Handling API Responses and Errors<br>4.4. Local Storage and Data Handling<br>4.4.1. Introduction to Local Storage<br>4.4.2. Using AsyncStorage<br>4.4.3. Storing and Retrieving Data<br>4.5. Application Testing in React Native<br>4.5.1. Introduction to Testing<br>4.5.2. Unit Testing Concepts<br>4.5.3. Debugging React Native Applications<br>4.6. Building and Deploying Mobile Applications<br>4.6.1. Building APK for Android<br>4.6.2. Preparing App for Release<br>4.6.3. Deploying Applications to App Stores |
| **Teaching Methodology** | Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions. |
| **Evaluation Scheme** | Internal Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br>External Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br><br>50% Internal Assessment:<br>- Attendance, class and home Assignments, Unit Tests<br>- Practical exam, viva-voce, Journal<br>50% External Assessment:<br>- Written Theory Examination<br>- Practical exam, Viva-Voce |
| **Reference Books** | 1. Flutter Complete Reference, Alberto Miola, Packt Publishing, 9781788832755, 2019<br>2. Flutter in Action, Eric Windmill, Manning Publications, 9781617296147, 2020<br>3. Beginning Flutter: A Hands-On Guide to App Development, Marco L. Napoli, Wiley Publishing, 9781119550822, 2019<br>4. Learning React Native (2nd Edition), Bonnie Eisenman, O'Reilly Media, 9781492056201, 2020 |

| | |
|---|---|
| | 5. React Native in Action, Nader Dabit, Manning Publications, 9781617294051, 2019<br>6. Flutter Recipes: Mobile Development Solutions for iOS and Android, Fu Cheng, Apress, 9781484249710, 2019<br>7. Fullstack React Native, Devin Abbott, Fullstack.io, 9780991344659, 2017<br>8. Flutter Cookbook: Over 100 Proven Techniques and Solutions for App Development with Flutter, Simone Alessandria, Packt Publishing, 9781789615807, 2019<br>9. Practical Flutter: Improve Your Mobile Development with Google's Latest Open-Source SDK, Frank Zammetti, Apress, 9781484251805, 2019<br>10. Hands-On React Native: Build Mobile Apps with JavaScript and React, Nader Dabit, Packt Publishing, 9781788991261, 2018 |

**[Subject code for Theory-2711000908022001]**

**[Subject code for Practical-2711000908022002]**

# Course 803 – AI Implementation using python

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** | | | | | | |
| **Semester** | **8th** | | | | | | |
| **NCrF Credit Level** | **6.0** | | | | | | |
| **Course Type** | **Major** | | | | | | |
| **Course Subtype** | **Nil** | | | | | | |
| **Subject Type** | **Intra-disciplinary** | | | | | | |
| **Course Code** | **803** | | | | | | |
| **Course Level** | **400-499 ( Advanced Level )** | | | | | | |
| **Course Title** | **AI Implementation using python** | | | | | | |
| **Credit** | **4** | **Theory:** | **2 hrs** | **Practical:** | **4 hrs** | **Total:** | **6 Hrs** |
| **Effective From** | **Academic Year :2026-27** | | | | | | |
| **Course Purpose** | The purpose of this course is to provide students with practical knowledge of implementing Artificial Intelligence models using Python. The course focuses on data preparation, machine learning model implementation, deep learning concepts, and computer vision applications. Students will gain hands-on experience with AI tools and libraries to build intelligent systems capable of solving real-world problems. | | | | | | |
| **Course Objectives** | 1. To introduce students to data preparation techniques and tools required for AI implementation.<br>2. To enable students to implement machine learning models using Python libraries.<br>3. To provide knowledge of deep learning concepts and neural network implementation.<br>4. To familiarize students with TensorFlow and Keras frameworks.<br>5. To enable students to develop computer vision applications using deep learning models. | | | | | | |
| **Pre-Requisite** | Python Programming, Data handling using Python (NumPy / Pandas), Basic Machine Learning concepts | | | | | | |
| **Course Outcomes (COs)** | **CO1**: Prepare and preprocess datasets using Python tools for AI applications.<br>**CO2**: Implement and evaluate machine learning models using Python libraries such as Scikit-learn.<br>**CO3**: Design and train Artificial Neural Network models using TensorFlow and Keras.<br>**CO4**: Apply deep learning techniques to solve computer vision problems such as image classification and object detection. | | | | | | |

**CO – PSO Mapping Table**

| | PSO1 | PSO2 | PSO3 | PSO4 | PSO5 | PSO6 | PSO7 | PSO8 |
|---|---|---|---|---|---|---|---|---|
| **CO1** | ✓ | ✓ | | ✓ | ✓ | | ✓ | |
| **CO2** | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | |
| **CO3** | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO4** | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |

| | |
|---|---|
| **Content** | **Unit 1: Python Tools and Data Preparation for AI**<br>1.1. Data Loading and Cleaning<br>1.1.1. Loading data with csv, excel, json<br>1.1.2. Data cleaning and preprocessing<br>1.2. Feature Scaling and encoding<br>1.3. Working with datasets from Kaggle / open data sources<br>1.4. Automated EDA<br><br>**Unit 2: AI Model Implementation using Python**<br>2.1. Model building workflow in AI projects<br>2.1.1. Train-test split and cross validation<br>2.1.2. Hyper parameter tuning<br>2.1.3. Model evaluation techniques<br>2.2. Implementation using Scikit-learn:<br>2.2.1 SVM, Random Forest, Gradient Boosting, Naïve Bayes<br>2.3. Model Evaluation<br>2.3.1 Accuracy, Precision, Recall, F1 Score<br>2.3.2 Confusion Matrix<br>2.3.3 ROC Curve<br><br>**Unit 3: Deep Learning using Python**<br>3.1. Introduction to Deep Learning<br>3.2. Artificial Neural Networks implementation<br>3.2.1. Structure of Artificial Neural Networks<br>3.2.2. Layers in Neural Networks (Input Layer, Hidden Layer, Output Layer)<br>3.2.3. Activation Functions: Sigmoid, ReLU, Tanh, Softmax<br>3.2.4. Forward and Backward Propagation<br>3.2.5. Overfitting and Underfitting<br>3.3. Introduction to TensorFlow and Keras<br>3.4. Techniques to improve model performance (Regularization, Dropout)<br><br>**Unit 4: Implementing Deep Learning in Computer Vision**<br>4.1. Introduction to Image Processing<br>4.1.1. Computer Vision and AI<br>4.1.2. Applications of Computer Vision<br>4.1.2.1. Face Recognition<br>4.1.2.2. Self-driving cars<br>4.1.2.3. Medical image analysis<br>4.1.2.4. Surveillance systems<br>4.2. Image Fundamentals<br>4.2.1. Digital Image representation<br>4.2.2. Pixels, resolution and color channels<br>4.2.3. Image formats and image datasets<br>4.2.4. Image preprocessing techniques: Resizing, Cropping, Normalization, Image augmentation.<br>4.3. Deep Learning for Computer Vision<br>4.3.1. Introduction to Convolutional Neural Networks (CNN)<br>4.3.2. CNN architecture (Convolution, Pooling, Fully Connected layers)<br>4.3.3. Image classification using CNN<br>4.3.4. Object detection using CNN |
| **Teaching Methodology** | Classroom Teaching, Hands-on Lab Sessions, Demonstrations, Assignments, and Discussions. |

| | |
|---|---|
| **Evaluation Scheme** | Internal Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br>External Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br><br>50% Internal Assessment:<br>- Attendance, class and home Assignments, Unit Tests<br>- Practical exam, viva-voce, Journal<br>50% External Assessment:<br>- Written Theory Examination<br>- Practical exam, Viva-Voce |
| **Reference Books** | 1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd Edition), Aurélien Géron, O'Reilly Media, 9781098125974, 2022<br>2. Python Machine Learning (3rd Edition), Sebastian Raschka; VahidMirjalili, Packt Publishing, 9781789955750, 2019<br>3. Deep Learning, Ian Goodfellow; YoshuaBengio; Aaron Courville, MIT Press, 9780262035613, 2016<br>4. Deep Learning with Python (2nd Edition), François Chollet, Manning Publications, 9781617296864, 2021<br>5. Machine Learning with Python Cookbook, Chris Albon, O'Reilly Media, 9781491989388, 2018<br>6. Practical Computer Vision with Python, Abhinav Dadhich, Packt Publishing, 9781788299336, 2018<br>7. Programming Computer Vision with Python, Jan Erik Solem, O'Reilly Media, 9781449316549, 2012<br>8. Introduction to Machine Learning with Python, Andreas C. Müller; Sarah Guido, O'Reilly Media, 9781449369415, 2016<br>9. Deep Learning for Computer Vision with Python, Adrian Rosebrock, PyImageSearch, 9781733684705, 2019<br>10. Applied Artificial Intelligence: A Handbook for Business Leaders, Mariya Yao; Adelyn Zhou; Marlene Jia, Topbots Publishing, 9780998289021, 2018 |

# Course 804: Automated Testing Framework

**[Subject code for *Theory*-2711000908033001]** ***[Subject code for Practical-2711000908033002]***

| | | | | | | |
|---|---|---|---|---|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** | | | | | |
| **Semester** | **8th** | | | | | |
| **NCrF Credit Level** | **6.0** | | | | | |
| **Course Type** | **Major** | | | | | |
| **Course Subtype** | **Nil** | | | | | |
| **Subject Type** | **Intra-disciplinary** | | | | | |
| **Course Code** | **804** | | | | | |
| **Course Level** | **400-499 ( Advanced Level )** | | | | | |
| **Course Title** | **Automated Testing Framework** | | | | | |
| **Credit** | **4** | **Theory:** | **2 hrs** | **Practical:** | **4 hrs** | **Total:** | **6 Hrs** |
| **Effective From** | **Academic Year :2026-27** | | | | | |

| | |
|---|---|
| **Course Purpose** | To introduce students to software testing concepts and the importance of testing in software development. To help students to understand the difference between manual and automated testing. To prepares students to become skilled in automated software testing using modern tools like Selenium**.** |
| **Course Objectives** | The objectives of this course is:<br>1. Students can understand the fundamentals of software testing and automation techniques.<br>2. Students can learn the use of Selenium tools and architecture for automated testing.<br>3. Students can gain practical knowledge of web element handling, navigation, and user interactions**.**<br>4. Students can also understand how to test real-time web applications including forms, login pages, and dynamic content.<br>5. Students can develop skills in debugging, reporting, and maintaining test scripts**.** |
| **Pre-requisite** | Concepts of Web Development and Operations |
| **Course outcome** | **CO1:** Describe the principles of automated testing, and differentiate between manual and automation tools and strategies.<br>**CO2:** Apply Selenium Web Driver using Python to automate real-time web applications including login, form submission, and navigation.<br>**CO3:** Analyze and handle dynamic elements, drop downs, alerts, pop ups, and mouse events in modern web applications.<br>**CO4:** Develop end-to-end test scripts using Selenium-Python integrated with Excel sheets and database operations.<br>**CO5:** Evaluate synchronization techniques and implement advanced testing features like Ajax handling, listeners, and reporting. |

**CO-PSO mapping**

| | PSO1 | PSO2 | PSO3 | PSO4 | PSO5 | PSO6 | PSO7 | PSO8 |
|---|---|---|---|---|---|---|---|---|
| **CO1** | ✓ | ✓ | | ✓ | ✓ | | ✓ | |
| **CO2** | | ✓ | | ✓ | ✓ | ✓ | ✓ | |
| **CO3** | | ✓ | | ✓ | ✓ | ✓ | ✓ | |
| **CO4** | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **CO5** | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**Course Content**

**Unit1:Introduction to Automated Testing and Selenium Tools**

1.1. Concepts of Software Testing
  1.1.1. Manual vs. Automated Testing –Pros & Cons
  1.1.2. Types of Tests Suitable for Automation
1.2. Selenium Overview
  1.2.1. Selenium IDE, RC, Web Driver, and Grid
  1.2.2. Installation of Selenium IDE, Fire Bug, Fire Path
  1.2.3. Selenium Architecture
  1.2.4. JSON Wire Protocol, Browser Drivers
  1.2.5. Selenium Client Libraries

**Unit 2:Selenium WebDriver with Python**

2.1. Introduction to Selenium-Python
  2.1.1. Installing Python-Selenium packages
  2.1.2. Basic Web Navigation using get()
  2.1.3. Interacting with Web Elements
2.2. Locating Web Elements
  2.2.1. find_element_by_id,name, xpath, tag_name, etc.
  2.2.2. Dynamic XPath & CSS Selectors
2.3. Action Chains for Advanced User Interactions
  2.3.1. Click, Click &Hold, Double Click, Drag & Drop
  2.3.2. Key Events: Key Down/Up, Perform, Pause, Release

**Unit3:Web Automation with Real-time Use Cases**

3.1. Selenium Web Driver with Different Browsers
  3.1.1. Chrome, Firefox, Edge, IE Configuration
3.2. Form Handling and Navigation
  3.2.1. Login Page Automation(e.g.,Facebook, Gmail)
  3.2.2. Methods: maximize_window(), get(), send_keys(), close()
3.3. Handling UI Components
  3.3.1. Drop-downs, Alerts, Pop ups, and Multi-window Handling
3.4. Mouse and Keyboard Interactions
  3.4.1. MouseHover, RightClick, DoubleClick, DragandDrop
3.5. Screen shot Capture
  3.5.1. Element Screenshot vs. FullPage Screenshot

**Unit4: Synchronization, Data Handling, and Backend Testing**

4.1. Selenium Waits
  4.1.1. Implicit, Explicit, and Fluent Waits
4.2. Apache POI for ExcelFile Integration
  4.2.1. Read/WriteExcel Data
4.3. Database Testing
  4.3.1. Using MySQL or DB2with Selenium
4.4. Handling Ajax Calls and Listeners
  4.4.1. JavaScript Executor for Dynamic Content
4.5. Test Reporting & Debugging Techniques(Optional Intro to TestNG or

| | |
|---|---|
| | PyTest) |
| **Reference Books** | 1. TheArtofSoftwareTesting,3rdEdition,GlenfordJ.Myers,Corey Sandler, Tom Badgett,<br>2. SoftwareTesting,2ndEdition,2005,RonPatton,SamsPublishing, ISBN-13: 978-0672327988,<br>3. Selenium with Python, Pallavi R Sharma, BPB Publication, ISBN-13:978-9389328813<br>4. PythonTestingwithSelenium,SujayRaghavendra,ISBN-13:978-1484262481<br>5. Selenium Web Driver, Rajeev Gupta,ISBN-13:978-9332526297<br>6. Guide To Test Automation Using Selenium, Gargand Aditya, McGraw Hill, ISBN: 9781259005930<br>7. Fundamentals of Database Systems, Ramez Elmasri, ISBN:9788131716250<br>8. Selenium Testing Tools Cookbook (2nd Edition), Unmesh Gundecha, Packt Publishing, 9781784392512, 2015<br>9. Mastering Selenium WebDriver (2nd Edition), Mark Collin, Packt Publishing, 9781784394356, 2015<br>10. Learning Selenium Testing Tools with Python, Pradeep Kumar Gaur; Gaurav Gupta, Packt Publishing, 9781783553600, 2015 |
| **Teaching Methodology** | Class Work, Discussion, Lab work, Self-Study, Seminars and/or Assignments |
| **Evaluation Method** | Internal Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br>External Assessment :25 Marks Theory + 25 Marks Practical = 50 Marks<br><br>50% Internal Assessment:<br>- Attendance, class and home Assignments, Unit Tests<br>- Practical exam, viva-voce, Journal<br>50% External Assessment:<br>- Written Theory Examination<br>- Practical exam, Viva-Voce |

*[Subject code-2711000908044001]*

# Course 805: On-the-Job Training

| | |
|---|---|
| **Program Name** | **Bachelor of Computer Science (Honours)** |
| **Semester** | **8th** |
| **NCrF Credit Level** | **6.0** |
| **Course Type** | **Major** |
| **Course Subtype** | **Nil** |
| **Subject Type** | **Intra-disciplinary** |
| **Course Code** | **805** |
| **Course Level** | **400-499 ( Advanced Level )** |
| **Course Title** | **On-the-Job Training** |
| **Credit** | **6** |
| **Effective From** | **Academic Year :2026-27** |
| **Course Purpose** | This course aims to provide students with first-hand exposure to the real-time working environment of the software and IT industry. Through structured industry visits and observational learning, students will understand the software development lifecycle, team structures, technology stacks, HR strategies, project methodologies, and company work culture. This immersion will bridge the gap between academic learning and industrial practices, enhancing the student's readiness for employment. The culminating activities—presentationandreport—helpstudentssynthesizetheirlearninganddevelopdocumentationandcommunication skills. |
| **Course Objectives** | The objectives of this course is:<br>• To provide students with real-world exposure to the IT industry through industry visits.<br>• To understand the working environment, roles, and organizational structure of software companies.<br>• To learn about the Software Development Life Cycle (SDLC) and industry practices like Agile.<br>• To observe and analyze team collaboration, tools, and technologies used in real projects**.**<br>• To develop skills in professional communication, observation, and data collection.<br>• To understand HR practices, company culture, and career growth opportunities**.**<br>• To enhance abilities in report writing, documentation, and presentation. |
| **Pre-requisite** | Students must have completed the seventh semester of the B.C.A. (honors) program and should possess fundamental knowledge of software Development and IT concepts. |

**Course outcome**

**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 proposes elf-improvement plans for better employability.

**Course Outcome / Program Specific Outcome mapping:**

| | PSO1 | PSO2 | PSO3 | PSO4 | PSO5 | PSO6 | PSO7 | PSO8 |
|---|---|---|---|---|---|---|---|---|
| **CO1** | ✓ | | ✓ | ✓ | | ✓ | | |
| **CO2** | | ✓ | | ✓ | ✓ | ✓ | ✓ | |
| **CO3** | ✓ | | ✓ | ✓ | | ✓ | | |
| **CO4** | ✓ | | ✓ | ✓ | ✓ | | | ✓ |
| **CO5** | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ |

**Phases and components of Course Outcomes:**

**1:OrientationandPreparationforIndustryVisit**

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:IndustryVisit and Observation**

2.1. Visiting Software Companies (Startups, Mid-size, or Large IT firms)
2.2. Understanding Software Development Life cycle (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,andOrganizational Analysis**

3.1. HR Functions: Hiring Strategies, Performance Appraisal, On boarding
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,andPresentation**

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

<table>
<tr>
<td>Project Evaluation:</td>
<td>50%Internal: Based on presentation on following components and viva-voce.<br>50%External: Based on presentation on following components and viva-voce.<br><br>
<table>
<tr><th>Component</th><th>Weightage</th></tr>
<tr><td>Participation& Conduct</td><td>10%</td></tr>
<tr><td>Observation Logs & Notes</td><td>10%</td></tr>
<tr><td>Final Report</td><td>40%</td></tr>
<tr><td>Presentation &Demo</td><td>30%</td></tr>
<tr><td>Reflective Discussion/Viva</td><td>10%</td></tr>
</table>
<br>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.</td>
</tr>
</table>