સૌરાષ્ટ્ર યુનિવર્સિટી

એકેડેમિક વિભાગ

યુનિવર્સિટી કેમ્પસ, યુનિવર્સિટી રોડ, રાજકોટ-૩૬૦૦૦૫

ફોન નં.(૦૨૮૧)૨૫૭૮૫૦૧ એક્સટે. નં.૨૦૨, ૩૦૪ ફેક્સ નં.(૦૨૮૧)૨૫૭૬૩૪૭ ઈ-મેઈલ : academic@sauuni.ac.in

નં.એકે/કોમ્પ્યુટર સાયન્સ/૨૬૧૯૬૫/૨૦૨૫ તા.૧૪/૧૦/૨૦૨૫

B.C.A., B.Sc.(IT) and B.Sc. CS

પરિપત્ર:-

સૌરાષ્ટ્ર યુનિવર્સિટીની કોમ્પ્યુટર સાયન્સ વિદ્યાશાખા હેઠળની સ્નાતક કક્ષાના બી.એસસી.(કોમ્પ્યુટર સાયન્સ) ના અભ્યાસક્રમ ચલાવતી સર્વે સંલગ્ન કોલેજોના આચાર્યશ્રીઓને આથી જાણ કરવામાં આવે છે કે, તા.૦૪/૦૯/૨૦૨૫ના રોજ B.C.A., B.Sc.(IT) and B.Sc. CS ના પ્રેક્ટીકલ પેપર પ્રશ્નના નિવારણ માટે કોમ્પ્યુટર સાયન્સ વિષયની અભ્યાસ સમિતિની એક સભાનું આયોજન કરેલ જેના કાર્યવાહી નોંધનાં ઠરાવ ક્રમાંક '૦૨' માં નીચે મુજબ ઠારાવવામાં આવેલ છે.

- B.C.A. અને B.Sc.(IT) ના અભ્યાસક્રમમાં SEE તરીકે જ્યાં પ્રેક્ટીકલ સબ્જેક્ટમાં થીયરી એક્ઝામ લેવાતી હતી તેની જગ્યાએ હવે ત્યાં પ્રેક્ટીકલ એક્ઝામ લેવાની રહેશે જેમાં ખાસ સેમેસ્ટર-1 થી સેમેસ્ટર-4 કોલેજ લેવલે અને સેમેસ્ટર-5 થી સેમેસ્ટર-6 માં યુનિવર્સિટી દ્વારા પ્રેક્ટીકલ એક્ઝામ લેવામાં આવશે
- જે વિદ્યાર્થીને પ્રેક્ટીકલ સબ્જેક્ટની થીયરી એક્ઝામમાં ATKT આવેલ હોય તેમને જૂની સ્કીમ પ્રમાણે જ થિયરીની એક્ઝામ આપવાની રહેશે જ્યાં સુધી ATKT રહેશે ત્યાં સુધી જુની સ્કીમ પ્રમાણે એમને પ્રેક્ટીકલ થિયરીની એકઝામ આપવાની રહેશે

  ઉપરોક્ત મુદ્દા ક્રમાંક (૧) અને (૨) માટે માન. કુલપતિશ્રીને અધિકાર મંડળોની બહાલીની અપેક્ષાએ મંજુરી આપવા ભલામણ કરવામાં આવેલ છે.
- ઉપર્યુક્ત બાબત માટે B.C.A. અને B.Sc.(IT) સત્ર (૧) થી (૬) ના અભ્યાસક્રમો રજુ કરવામાં આવ્યા અને તેને માન. કુલપતિશ્રી સાહેબને અધિકાર મંડળોની બહાલીની અપેક્ષાએ મંજુરી આપવા ભલામણ કરવામાં આવેલ છે.
- ડીનશ્રી, તથા ચેરપર્સનશ્રી, કોમ્પ્યુટર સાયન્સ વિષયની અભ્યાસ સમિતિની તેમજ કોમ્પ્યુટર સાયન્સ વિષયની અભ્યાસ સમિતિની તા.૦૪/૦૯/૨૦૨૫ની કાર્યવાહી નોંધનાં ઠરાવ ક્રમાંક '૦૨' અન્વયે સત્ર ૧ થી ૬ અભ્યાસક્રમ અધિકાર મંડળોની બહાલીની અપેક્ષા મંજુરી આપવા માન.કુલપતિ સાહેબને ભલામણ કરેલ જે માન.કુલપતિશ્રીએ મંજુર કરેલ છે. જેથી સંબંધિત તમામે તે મુજબ તેની ચુસ્તપણે અમલવારી કરવી.

(મુસદ્દો કુલસચિવશ્રીએ મંજુર કરેલ છે.)

સહી/-
(ડૉ.આર.જી.પરમાર)
કુલસચિવ

બિડાણ:- ઉક્ત અભ્યાસક્રમ (સોફ્ટ કોપી)

રવાના કર્યું
14/10/25
એકેડેમિક ઓફીસર

પ્રતિ,

(૧) કોમ્પ્યુટર વિદ્યાશાખા હેઠળની B.C.A., B.Sc.(IT) and B.Sc. CS વિષય ચલાવતી સ્નાતક કક્ષાની સર્વે સંલગ્ન કોલેજોના આચાર્યશ્રીઓ તરફ
(૨) કોમ્પ્યુટર સાયન્સની વિષયની અભ્યાસ સમિતિના સર્વે સભ્યશ્રીઓ
(૩) ડીનશ્રી, કોમ્પ્યુટર સાયન્સ વિદ્યાશાખા

નકલ જાણ અર્થે રવાના:-

૧. માન.કુલપતિશ્રી/કુલસચિવશ્રીના અંગત સચિવ

નકલ રવાના (યોગ્ય કાર્યવાહી અર્થે):-

૧. પરીક્ષા વિભાગ ૨. પી.જી.ટી.આર.વિભાગ ૩. જોડાણ વિભાગ

Saurashtra University

Rajkot – Bharat

Curriculum
of
4 Year UG Programme

Bachelor of Computer Application (Honours)

&

Bachelor of Computer Application
(Honours with Research)

(Semester - 5 and Semester – 6)

To be effective from June – 2025

B.C.A. (Honours) & B.C.A. (Honours with Research)
(Semester - 5 and Semester - 6)
To be effective from June – 2025
Saurashtra University

SEE (50 marks) Paper setting guide lines for all the semester

Question Paper contains 5 questions (each of 10 marks). Every question will be asked from respective unit as specified in the syllabus of each course. (i.e., Question-1 from Unit No.1 and remaining questions from their respective units). Every question is divided in three parts like (a), (b) and (c). Part (a) contains three objective type questions (not MCQ) like definition, reason, answer in one line, answer in one word etc., each of one mark and no internal option. Part (b) contains two questions each of two marks and student will attempt any one out of two. Part (c) contains two questions each of five marks and student will attempt any one out of two.

Saurashtra University **BCA Semester- 1/2/3/4/5/6**					
Time: 02:00					**Total marks: 50**
Q. 1	(a)	Attempt the following		Unit-1	03
		(1)			
		(2)			
		(3)			
	(b)	Attempt any one of the following			02
		(1)			
		(2)			
	(c)	Attempt any one of the following			05
		(1)			
		(2)			
Q. 2	(a)	Attempt the following		Unit-2	03
		(1)			
		(2)			
		(3)			
	(b)	Attempt any one of the following			02
		(1)			
		(2)			
	(c)	Attempt any one of the following			05
		(1)			
		(2)			
Q. 3	(a)	Attempt the following		Unit-3	03
		(1)			
		(2)			
		(3)			
	(b)	Attempt any one of the following			02
		(1)			
		(2)			
	(c)	Attempt any one of the following			05
		(1)			
		(2)			
Q. 4	(a)	Attempt the following		Unit-4	03
		(1)			

		(2)			
		(3)			
	(b)	Attempt any one of the following			02
		(1)			
		(2)			
	(c)	Attempt any one of the following			05
		(1)			
		(2)			
Q. 5	(a)	Attempt the following			03
		(1)			
		(2)			
		(3)			
	(b)	Attempt any one of the following		Unit-5	02
		(1)			
		(2)			
	(c)	Attempt any one of the following			05
		(1)			
		(2)			

SEE (25 marks) Paper setting guide lines for all the semester

Question Paper contains 3 questions (Q.1 – 10 marks, Q.2 – 10 marks and Q.3- 05 marks). Q.1 is from unit-1, contains four questions each of five marks and student will attempt any two out of four. Q.2 is from unit-2, contains four questions each of five marks and student will attempt any two out of four. Q.3 is from unit-3, contains two questions each of five marks and student will attempt any one out of two.

Saurashtra University
BCA Semester- 1/2/3/4/5/6

Time: 01:00 **Total marks: 25**

Q. 1	Attempt any two of the following			10
	(1)			
	(2)		Unit-1	
	(3)			
	(4)			
Q. 2	Attempt any two of the following			10
	(1)			
	(2)		Unit-2	
	(3)			
	(4)			
Q. 3	Attempt any one of the following			05
	(1)		Unit-3	
	(2)			

B.C.A. (Honours) & B.C.A. (Honours with Research)
(Semester - 5 and Semester - 6)
To be effective from June – 2025
Saurashtra University

BCA SEM 5						
Sr. No	**Type Of Course**	**Subject**	**Credit**	**CCE**	**SEE**	**Total**
1	Major-11	**CS –29:** Advance Java Programming(J2EE)	4	50	50	100
2	Major-12	**CS –30:** Programming in Python	4	50	50	100
3	Major-13	**CS –31:** Cyber Security	4	50	50	100
4	Minor-04	**CS –32:** Mini Project	4	50	50	100
5	Minor-05	**CS –33:** Practical Based on CS-29 and CS-30	4	50	50	100
6	SEC-05	**CS –34:** Introduction to AI	2	25	25	50
		Total credit	22			

BCA SEM 6						
Sr. No	**Type Of Course**	**Subject**	**Credit**	**CCE**	**SEE**	**Total**
1	Major-14	**CS –35 :** Mobile Application Development in Android using Kotlin	4	50	50	100
2	Major-15	**CS –36 :** Programming with ASP.NET	4	50	50	100
3	Major-16	**CS –37 :** Machine Learning with Python	4	50	50	100
4	Minor-06	**CS –38 :** Practical Based on CS-35, CS-36, CS-37	4	50	50	100
5	AEC-05	**CS –39 :** Introduction of Cloud Computing	2	25	25	50
6	SEC06 (Internship)	**CS –40 :** Internship	4	0	100	100
		Total Credit	22			

BCA SEM 5						
Sr. No	**Type Of Course**	**Subject**	**Credit**	**CCE**	**SEE**	**Total**
1	Major-11	**CS –29:** Advance Java Programming(J2EE)	4	50	50	100
2	Major-12	**CS –30:** Programming in Python	4	50	50	100
3	Major-13	**CS –31:** Cyber Security	4	50	50	100
4	Minor-04	**CS –32:** Mini Project	4	50	50	100
5	Minor-05	**CS –33:** Practical Based on CS-29 and CS-30	4	50	50	100
6	SEC-05	**CS –34:** Introduction to AI	2	25	25	50

<table>
<tr><th colspan="3">CS-29: Advance Java Programming (J2EE)</th></tr>
<tr><td colspan="3">Objectives:
Gain a deep understanding of the principles of J2EE architecture, including servlets, JSP.
Proficiency in frameworks and technologies like spring framework, hibernate, spring boot.
Learn about the Model-View-Controller (MVC) design pattern and its application in J2EE development.
Prerequisites:
Core Java Knowledge</td></tr>
<tr><th>Unit No.</th><th>Topic</th><th>Detail</th></tr>
<tr><td rowspan="2">1</td><td rowspan="2">Introduction to J2EE and JDBC</td><td>● Introduction to J2EE
● Enterprise Architecture Styles:
▪ Two-Tier Architecture
▪ Three-Tier Architecture
▪ N-Tier Architecture
● Enterprise Architecture
● The J2EE Platform
● Introduction to J2EE APIs (Servlet, JSP, EJB, JMS, JavaMail, JSF, JNDI)
● Introduction to Containers
● Tomcat as a Web Container</td></tr>
<tr><td>● JDBC Architecture,
● Types of JDBC Drivers,
● Introduction to major JDBC Classes and Interface,
● Creating simple JDBC Application,
● Types of Statement (Statement Interface, PreparedStatement, CallableStatement),
● Creating CRUD Application</td></tr>
<tr><td>2</td><td>Servlet</td><td>● Servlet Introduction
● Architecture of a Servlet
● Servlet API (Javax.servlet and Javax.servlet.http)
● Servlet Life Cycle
● Servlet Configuration with Deployment Descriptor
● Developing and Deploying Servlets
● Handling Servlet Requests and Responses
● Reading Initialization Parameters
● Session Tracking Approaches (URL Rewriting, Hidden Form Fields, Cookies, Session API)</td></tr>
</table>

3	JSP	● Introduction to JSP and JSP Basics ● JSP vs. Servlet ● JSP Architecture ● Life cycle of JSP ● JSP Elements: ▪ Directives Elements (page, include, taglib) ▪ Scripting Elements (Declaration, scriptlet, expression) ▪ Action Elements (jsp:param, jsp:include, jsp:Forward, jsp:plugin, jsp:useBean, jsp:setAttribute, jsp:getAttribute) ● JSP Implicit Objects (request, response, out, session, application, pagecontext) ● JSP Scope ● Including and Forwarding from JSP Pages ▪ include Action ▪ forward Action ● Working with Session & Cookie in JSP ● Error Handling and Exception Handling with JSP ● JSP EL (Expression Language), JSP Standard Tag Libraries (JSTL)
4	EJB, Introduction of MVC Architecture, Hibernate	● Introduction to EJB ● Types of EJB
		● Introduction to MVC ● Implementation of MVC Architecture
		● Introduction to Hibernate ● Features of Hibernate ● Exploring Hibernate Architecture ● Object Relation Mapping (ORM) with Hibernate ● Hibernate Configuration file ● Hibernate Mapping file ● Hibernate Annotation ● Hibernate Query Language (HQL) ● Hibernate Sessions
5	Introduction to Spring Framework & Spring Boot	**Introduction to Spring Framework** ● What is Spring Framework? ● Importance and Benefits of Spring ● Spring Architecture Overview (Core Container, Modules) **Core Concepts of Spring** ● Dependency Injection (DI) and Inversion of Control (IoC) ● Bean Lifecycle in Spring ● Spring ApplicationContext and BeanFactory **Spring MVC Basics** ● Overview of MVC Pattern ● Spring MVC Architecture and Request Flow ● Controllers, Models, Views ● Simple Spring MVC Application Example

		Aspect-Oriented Programming (AOP) in Spring • Introduction to AOP • Concepts: Advice, Join Point, Pointcut, Aspect • Simple Logging Aspect Example **Spring Boot Fundamentals** • What is Spring Boot? • Features and Advantages of Spring Boot • Architecture of Spring Boot • Auto-Configuration and Starter Dependencies • Embedded Servers (Tomcat, Jetty) • Important Spring Boot Annotations (@SpringBootApplication, @RestController, @RequestMapping) **Spring Boot Data Access** • Introduction to Spring Boot JDBC • Using JdbcTemplate for database operations • Overview of Spring Boot Data JPA • Basic CRUD operations with Spring Boot Data JPA and Hibernate **Exception Handling in Spring Boot MVC** • Handling exceptions using @ExceptionHandler • Using @ControllerAdvice for global exception handling

Reference Books:

(1) Java Complete Reference 11th Edition - Herbert Schildt, Oracle Press
(2) Java Server Programming For Professionals, Ivan Bayross, Sharanam Shah – Shroff publication
(3) Developing Java Servlets – Techmedia
(4) JSP Beginner's Guide – Tata McGraw Hill by Gary Bolling, Bharathi Nataragan
(5) Spring and Hibernate, K. Santosh Kumar, - Tata McGraw-Hill
(6) Hibernate Made Easy: Simplified Data Persistence with Hibernate and JPA (Java Persistence API) Annotations by Cameron Wallace McKenzie, Kerri Sheehan
(7) Spring Framework: A Step by Step Approach for Learning Spring Framework – Create Space Independent Publishing Platform
(8) Beginning Hibernate Second Edition By Jeff Linwood, Dave Minte – Apress

Course Outcomes:

- Students should gain a comprehensive understanding of the architecture of J2EE, including its various tiers such as presentation, business logic, and data tiers.
- Students should be able to develop enterprise applications using J2EE technologies, including the ability to design and implement user interfaces, business logic, and data access layers.
- Students should be proficient in using various Java EE APIs for developing enterprise applications, including Servlets, JavaServer Pages (JSP).
- Understands and implements JSP and frameworks like Spring, Spring Boot etc.
- Understand and apply the concepts of MVC and tag libraries.

<table>
<tr><th colspan="3">CS-30: Programming in Python</th></tr>
<tr><td colspan="3">Objectives:
Understanding basic syntax of python and emphasize the importance of writing clear and concise code documentation and comments.
Familiarize students with Python’s style guide and best practices for writing Pythonic code.
Familiarize students with built-in data structures in Python such as lists, tuples, dictionaries etc.
Prerequisites:
Basic Computer Skills.
Fundamental Programming Concepts.
Problem-Solving Skills.</td></tr>
<tr><th>Unit No.</th><th>Topic</th><th>Detail</th></tr>
<tr><td>1</td><td>Introduction to Python</td><td>The basic elements of Python
Branching programs
Strings and Input
Iteration
Functions and Scoping, Specifications, Recursion
Global variables, Modules, Files
Tuples, Lists and Mutability
Functions as Objects, Strings
Tuples and Lists, Dictionaries</td></tr>
<tr><td>2</td><td>OOP using Python</td><td>Handling exceptions,
Exceptions as a control flow mechanism,
Assertions, Abstract Data Types and Classes,
Inheritance,
Encapsulation and information hiding,
Search Algorithms, Sorting Algorithms,
Hashtables</td></tr>
<tr><td>3</td><td>Plotting using PyLab</td><td>Plotting using PyLab,
Plotting mortgages and extended examples,
Fibonacci sequence revisited, Dynamic
programming and the 0/1 Knapsack algorithm,
Dynamic programming and divide and conquer</td></tr>
</table>

4	Network Programming and GUI using Python	• Network Programming: ▪ Protocol, Sockets, ▪ Knowing IP Address, ▪ URL, Reading the Source Code of a Web Page, ▪ Downloading a Web Page from Internet, ▪ Downloading an Image from Internet, ▪ A TCP/IP Server, A TCP/IP Client, ▪ A UDP Server, A UDP Client, ▪ File Server, File Client, ▪ Two-Way Communication between Server and Client, ▪ Sending a Simple Mail. • GUI Programming: ▪ Event-driven programming paradigm; ▪ creating simple GUI; ▪ buttons, labels, entry fields, dialogs; ▪ widget attributes - sizes, fonts, colors, treeview, layouts, nested frames
5	Connecting with Database	• Verifying the MySQL dB Interface Installation, • Working with MySQL Database, • Using MySQL from Python, • Retrieving All Rows from a Table, • Inserting Rows into a Table, • Deleting Rows from a Table, • Updating Rows in a Table, • Creating Database Tables through Python

ReferenceBooks:

- "Core Python Programming" by Dr.R. NageswaraRao– 2017 Edition, Dreamtech Press
- JohnVGuttag."Introduction to Computation and Programming Using Python", Prentice Hall of India
- Robert Sedgewick,KevinWayne,Robert Dondero, Introduction to Programming in python, Pearson
- WesleyJ Chun,CorePython ApplicationsProgramming,3rd Edition.Pearson
- Michael Bowles, Machine Leaning in Python, Esssential techniques for predictive analysis, Wiley

Course Outcomes:

- Understand the concept of programming with Python
- Understand the OOP using Python
- Implementing the plotting using PyLab
- Understand the Network Programming and GUI
- Understand and Implement database connectivity

<table>
<tr><th colspan="3">CS-31: Cyber Security</th></tr>
<tr><td colspan="3">Objectives:
Learn the foundations of Cyber Security and threat landscape.
To equip students with the technical knowledge and skills needed to protect and defend against cyber threats.
To expose students to governance, regulatory, legal, economic, environmental, social and ethical contexts of cyber security.
To develop skills in students that can help them plan, implement, and monitor cyber security mechanisms to ensure the protection of information technology assets.
Prerequisites:
Basic Computer Skills.
OS and Programming knowledge, Networking Fundamentals
Critical Thinking and Problem-Solving Skills</td></tr>
<tr><th>Unit No.</th><th>Topic</th><th>Detail</th></tr>
<tr><td>1</td><td>Introduction to Cyber Security</td><td>Defining Cyberspace and Overview of Computer and Web-technology
Architecture of cyberspace,
Communication and web technology,
Internet, World wide web,
Advent of internet,
Internet infrastructure for data transfer and governance,
Internet society,
Regulation of cyberspace
Concept of cyber security
Issues and challenges of cyber security</td></tr>
<tr><td>2</td><td>Cyber Crime and Cyber law</td><td>Classification of cyber crimes
Common cyber crimes
cyber crime targeting computers and mobiles
Cyber crime against women and children
Financial frauds
Social engineering attacks
Malware and ransomware attacks
Zero day and zero click attacks
Cybercriminals modus-operandi
Reporting of cyber crimes
Remedial and mitigation measures
Legal perspective of cyber crime
IT Act 2000 and its amendments
Cyber crime and offences
Organisations dealing with Cyber crime and Cyber Security in India</td></tr>
</table>

		• Case studies
3	Social Media Overview and Security	• Introduction to Social networks • Types of Social media • Social media platforms • Social media monitoring • Hashtag • Viral content • Social media marketing • Social media privacy • Challenges, opportunities and pitfalls in Online Social Network • Security issues related to social media • Flagging and reporting of inappropriate content • Laws regarding posting of inappropriate content • Best practices for the use of Social media • Case studies
4	E-commerce and Digital Payments	• Definition of E-Commerce • Main components of E-Commerce • Elements of E-Commerce security • E-Commerce threats • E-Commerce security best practices • Introduction to digital payments • Components of digital payment and stake holders • Modes of digital payments: o Banking Cards o Unified Payment Interface (UPI) o e-Wallets o Unstructured Supplementary Service Data (USSD) o Aadhar enabled payments • Digital payments related common frauds and preventive measures • RBI guidelines on digital payments and customer protection in unauthorized banking transactions • Relevant provisions of Payment Settlement Act, 2007.
5	Digital Devices Security, Tools and Technologies for Cyber Security	• End Point device and Mobile Phone security • Password policy • Security patch management • Data backup • Downloading and management of third party software • Device security policy • Cyber Security best practices • Significance of host firewall and Anti-virus • Management of host firewall and Anti-virus

		• Wi-Fi security • Configuration of basic security policy and permissions

Reference Books:

- Cyber Crime Impact in the New Millenium, by R. C. Mishra, Auther Press. Edition 2010.
- Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Sumit Belapure and nina Godbole, Wiley India Pvt. Ltd. (First Edition, 2011)
- Security in the Digital Age: Social Media Security Threats and Vulnerabilities by Henry A. Oliver, Create Space Independent Publishing Platform (Pearson, 13th November, 2001)
- Electronic Commerce by Elias M. Awad, Prentice Hall of India Pvt. Ltd.
- Cyber Laws: Intellectual Property & E-Commerce Security by Kumar K, Dominant Publishers
- Network Security Bible, Eric Cole, Ronald Krutz, James W. Conley, 2nd Edition, Wiley India Pvt. Ltd.
- Fundamentals of Network Security by E. Maiwald, McGraw Hill.

Course Outcomes:

- After completion of this module, students would be able to understand the concept of Cyber security and issues and challenges associated with it.
- Understand the cyber crimes, their nature, legal remedies and as to how report the crimes through available platforms and procedures.
- Able to appreciate various privacy and security concerns on online Social media and understand the reporting procedure of inappropriate content, underlying legal aspects and best practices for the use of Social media platforms.
- Understand the basic concepts related to E-commerce and digital payments.
- Familiar with various digital payment modes and related cyber security aspects, RBI guidelines and preventive measures against digital payment frauds.
- Understand the basic security aspects related to Computer and Mobiles.
- Able to use basic tools and technologies to protect their devices.

<table>
<tr><th colspan="4">CS-32: Mini Project</th></tr>
<tr><td colspan="4">Objectives:
• To gain hands-on experience in using programming languages, tools, and technologies.
• To understand and implement the complete Software Development Life Cycle (SDLC).
• To enhance problem-solving and analytical skills by designing solutions to real-life problems.
• To improve project planning and time management abilities.
• To practice documentation and reporting skills.</td></tr>
<tr><th>No.</th><th>Topic</th><th colspan="2">Guidelines for Mini Project</th></tr>
<tr><td>1</td><td>General</td><td colspan="2">• Choose a project topic
• Projects can be individual or group-based (2 Students)
• Submit a project proposal with the title, objective, scope ,tools/ technologies to be used, and team members.
• Get the proposal approved by the project guide/faculty coordinator.
• Use any suitable programming language or platform
• Project must be under the supervision of college faculties.</td></tr>
<tr><td>2</td><td>Documentation</td><td colspan="2">• Maintain a project report including the following:
• Title Page
• Certificate
• Acknowledgment
• Table of Contents
• Introduction & Objective
• System Analysis (Problem Definition, Feasibility Study)
• System Design (DFD, ER Diagram, etc.)
• Data Dictionary
• Implementation (Screenshots)
• Testing (Test cases and results)
• Conclusion & Future Scope</td></tr>
<tr><td>3</td><td>Submission</td><td colspan="2">• Submit a soft copy (CD/Pen drive) and hard copy of the project report for SEE. Project report must be printed on both side of the page.</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Assessment Criteria</td><td>CCE –Continuous and comprehensive evaluation done by institute.</td><td>50 Marks</td></tr>
<tr><td>SEE –Exam is conducted by Saurashtra University using external examiner (examiner will evaluate the executable project, hard copy of the project and take the viva voce)</td><td>50 Marks</td></tr>
</table>

CS-33: Practical Based on CS – 29 and CS – 30	
CCE- Continuous and comprehensive Evaluation as follow • The continuous Comprehensive Evaluation (CCE) for each subject will be conducted by the teacher of that subject. The teacher will decide how the evaluation will be done. Usually CCE includes things like class participation, case studies and presentation, assignments, tutorials, small test (announced or surprised), quizzes and attendance or a mix of these. • Students must submit their work for internal evaluation on time to time. • Another part of CCE is the mid-term exam, which is compulsory for all students. This exam will be conducted internally by the college.	50 Marks
SEE – Semester End Examination as per the following • Practical exams may be scheduled before or after the theory examinations. Exam is conducted by Saurashtra University using external examiner (3 hours duration) • Students must prepare a practical notebook/book for the final practical examination. (The practical book serves as a record of all practical work, observations, procedures and results performed during the semester in lab. It is essential for evaluation during the final practical examination)	50 Marks

CS-33: Practical Based on CS – 29 and CS – 30	**Total Marks - 100**	
Topics	**CCE**	**SEE**
CS-29: Advance Java Programming (J2EE)	**25**	**25**
CS-30: Programming in Python	**25**	**25**

CS-34: Introduction to AI

Objectives:

- Develop a comprehensive understanding of the fundamental concepts and applications of Artificial Intelligence.
- Gain knowledge of the major techniques and technologies used in Machine Learning and their applications in various domains.
- Develop an understanding of Natural Language Processing and its applications in fields such as chatbots, sentiment analysis, and language translation.
- Explore the applications and techniques of Computer Vision in real-world scenarios and understand the ethical considerations related to its use.
- Stay up-to-date with emerging trends and advancements in AI, and understand their implications for society and the workforce

Prerequisites:

-

Unit No.	Topic	Detail
1	Introduction to AI	• Definition of AI • Brief History of AI • Applications of AI • Ethical considerations in AI • Overview of AI Technologies and techniques
2	Computer Vision	• Introduction • Basic techniques of Computer Vision • Applications of Computer Vision • Computer Vision Libraries and Tools • Ethical Considerations in Computer Vision
3	Emerging Trends in AI	• Advanced AI technologies and techniques • AI and IOT • AI and Robotics • Future directions of AI research and development • Implications of AI for society and the workforce

ReferenceBooks:

- Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
- Shane, M. (2018). Artificial intelligence and ethics. Morgan & Claypool Publishers.
- Russell, S. J., & Norvig, P. (2020). Artificial intelligence: A modern approach. Pearson.

Course Outcomes:

- Students will be able to define Artificial Intelligence, describe its history and applications,

and analyze ethical considerations related to AI.

- Students will be able to understand the basics of Machine Learning, including the different types of algorithms, data preparation, and processing. They will also be able to identify successful Machine Learning projects.
- Students will be able to identify the different techniques used in Computer Vision, understand the applications of Computer Vision, and identify the ethical on side rations related to Computer Vision.
- Students will be able to identify emerging trends in Artificial Intelligence, including advanced AI technologies and techniques, AI and IoT, AI and Robotics, and future directions of AI research and development. They will also be able to analyze the implications of AI for society and the workforce.

BCA SEM 6						
Sr. No	**Type Of Course**	**Subject**	**Credit**	**CCE**	**SEE**	**Total**
1	Major-14	**CS –35 :** Mobile Application Development in Android using Kotlin	4	50	50	100
2	Major-15	**CS –36 :** Programming with ASP.NET	4	50	50	100
3	Major-16	**CS –37 :** Machine Learning with Python	4	50	50	100
4	Minor-06	**CS –38 :** Practical Based on CS-35, CS-36, CS-37	4	50	50	100
5	AEC	**CS –39 :** Introduction of Cloud Computing	2	25	25	50
6	SEC (Internship)	**CS –40 :** Internship	4	0	100	100

CS-35: Mobile Application Development in Android using Kotlin

Objectives:

- Understanding Android Development.
- Familiarize students with the Kotlin Programming Language
- Gain necessary knowledge and skills to develop high-quality Android applications using Kotlin.

Prerequisites:

- Basic Programming Knowledge
- Basic Understanding of Java and XML
- Knowledge of OOP Concepts

Unit No.	Topic	Detail
1	Introduction to Kotlin Programming	• Basics of Kotlin, Operations and Priorities, • Decision Making • Loop Control, Data Structures(Collections), • Functions • Object Oriented Programming: Inheritance abstract, interface, super and this, visibility modifiers.
2	Introduction to Android & Android Application Design	• The Open Handset Alliance, The Android Platform, Android SDK • Building a sample Android application • Anatomy of an Android applications, Android terminologies • Application Context, Activities, Services, Intents • Receiving and Broadcasting Intents • Android Manifest File and its common settings • Using Intent Filter, Permissions • Managing Application resources in a hierarchy • Working with different types of resources

3	Android User Interface Design	• User Interface Screen elements
		  • Button, EditText, TextView, DatePicker, TimePicker, ProgressBar, ListView, GridView, RadioGroup, ImageButton, Fragement
		• Designing User Interfaces with Layouts
		  • Relative Layout, Linear Layout, Table Layout etc
		• Dialogs
		• Drawing and Working with Animation
		  • Frame By Frame Animation
		• Twined Animation

4	Database Connectivity Using SQLite and Content Provider	• Using Android Data and Storage APIs • Managing data using SQLite • Sharing Data Between Applications with Content Providers

5	Location Based Services (LBS), Common Android API, Notifications, Services, Deployment of applications	• Using Global Positioning Services (GPS) • Geo coding Locations • Mapping Locations • Many more with location based services • Android networking API • Android web API • Android telephony API • Notifying the user, Notifying with the status bar • Vibrating the phone • Blinking the lights • Customizing the notifications Services • Application development using JSON in MySQL • Publish android application

Notes: Android application must be developed using ANDROID STUDIO 4.0

Reference Books:

(1) Learn Android Studio3 with Kotlin–TegHagos–Apress–2019
(2) Head first Kotlin, A Brain Friendly Guide–Dawn Griffiths, David Griffiths–Orilly–2019
(3) Professional Android2 Application Development Reto Meier,Wiley India Pvt Ltd (2011)
(4) Beginning Android Mark L Murphy, Wiley India Pvt Ltd
(5) Android Developer Fundamental Course – Practical Book – 2018

Course Outcomes:

- Understand the basic of KOTLIN programming.
- Understand the basic of Android and Android Application Design.
- Understand the different user interface elements and develop application with those widgets.
- Understand, apply and develop application with SQLite and Content Providers.
- Understand, apply and develop application with Location based services, notification services.

CS-36: Programming with ASP.NET

Objectives:

- Familiarize students with the basic concepts of ASP.NET
- Explore the different approaches to building web applications in ASP.NET
- Learn to integrate databases with ASP.NET applications using technologies like ADO.NET

Prerequisites:

- Basic Programming Knowledge
- Basic Understanding of HTML, CSS, OOP Concepts and C#.NET
- Having a general understanding of web development concepts such as client-server architecture, web servers etc.

Unit No.	Topic	Detail

1	Framework and Web Contents Validation Controls	• Overview of Asp.NET Framework
		• Client Server Architecture
		• Application Web Servers
		• Types of Files in Asp.NET
		• Types of controls in Asp.NET
		• Page Architecture
		• Web form
		• Introduction to standard Controls (Buttons, Textbox, Checkbox, Lable, Panel, Listbox, Dropdownlist etc.)
		• Running an Asp.Net Application, File Upload Control
		• What is Validation?
		  ◦ Client Side Validation
		  ◦ Server Side Validation
		• Types (RequieredField Validator, Range Validator, CompareField Validator, RegularExpression Validator, Custom Validator, ValidationSummery Control)

2	State Management	• What is State? • Why it is required in Asp.Net? • Client Side State Management • Server Side State Management • Various State Management Techniques (View State, Query String, Cookie, Session State,Application State)
3	ADO .NET and Database	• Architecture of ADO.NET • ADO.NET Classes for Connected and Disconnected Architecture (Connection, Command,DataReader, DataAdapter, DataSet, DataColumn, DataRow, DataConstraints, DataView etc.) • TheGridview Control, TheRepeater Control • Binding Datato DataBound Controls, • Diplaying Data in a webpage using SQLDataSource Control • DataBinding Expressions

4	Master Pages and Theme Caching, Application Pages and Data	• What is MasterPage?
		• Requirement Of a Master Page in an Asp.NET application
		• Designing Website with Master Page, Theme and CSS
		• Overview of Caching
		  o PageOutput Caching
		  o Partial Page Caching, Absolute Cache Expiration
		  o Sliding Cache Expiration
		  o Data Caching

5	Working with XML ASP.NET Application Configuration and Deployment of Application	• Reading Datasets FromXML • Writing DataSets With XML • WebServices (Introduction, HTTP, SOAP, UDDI,XML, Creating a Web Service, Consuming a Web Service) • Introduction To Web.Config • Common Configuration Sections • AppSettings • Tracing • Custom Errors • Authentication And Authorization • Deployment of Application in web server

Reference Books:

- ASP.NET - Unleashed
- ASP.NET – Wrox Publication
- Pro ASP.NET Core MVC 2 Book by Adam Freeman
- Introduction to ASP.NET Web Programming using the Razor Syntax (C#) by Tom FitzMacken

Course Outcomes:

- Understand the ASP.NET framework and different controls.
- Understand form validation, apply form validation control also understand state management.
- Understand ADO .NET architecture and developing application with LINQ.
- Understand and apply concept of Master Page, CSS & Theme.
- Understand configuration of application with XML.

CS-37: Machine Learning with Python

Objectives:

- To Understand and develop model of ML with Python.
- Apply ML techniques to real-world data sets and problems.
- Learn how to deploy machine learning models into production environments.

Prerequisites:

- Basic Understanding of Python Programming.

Unit No.	Topic	Detail
1	Introduction to Machine Learning	• Introduction to ML, Relation of ML with AI and DL, Defining Machine Learning, How machines learn, types of machine learning: supervised learning, unsupervised learning, reinforcement learning, applications of machine learning.
2	Supervised Learning	• Regression: Pre-processing data using different techniques – mean removal, scaling, normalization, binarization, label encoding, linear regression, case study implementation using Python • Classification: Building simple classifier, logistic regression classifier, Naïve bayes classifier, training and testing dataset, accuracy using cross-validation, visualizing confusion matrix, extracting the performance report. • Predictive Modeling: Building linear and non-linear classifier using Support Vector Machine (SVM), extracting confidence measurements, Case study implementation using Python.
3	Unsupervised Learning	• Clustering: Data using k-means clustering, compressing image using vector quantization, mean shift clustering model, agglomerative clustering, case study implementation using Python.
4	Natural Language Processing	• Natural Language Processing: o pre-processing data, o stemming data, o using lemmatization, o diving chunks, o text classifier, o case study implementation using Python.
5	Computer Vision with OpenCV	• Object Detection: o Detecting and tracking objects using Haar cascades from images and videos o Detecting face, eyes, mouth, nose, pupils

Reference Books:

▪ "Machine Learning" by Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das - Pearson

- "Python Machine Learning Cookbook" by Prateek Joshi – PACKT Publishing – 2016 Edition.
- "OpenCV: Computer Vision Projects with Python – Learning Path" by Joseph howse, Prateek Joshi, Michael Beyeler – PACKT Publishing – 2016 Edition.

Course Outcomes:

- To define and explain machine learning and its relation with AI and DL along with types of ML.
- To determine regression or classification supervised learning method of ML to any real-life application and estimate accuracy of the model.
- To be able to contrast various unsupervised learning methods and solve any real0life situation using ML and estimate accuracy of the model.
- To solve any fundamental text-processing.
- To construct a model to detect object from it.

CS-38: Practical Based on CS – 35, CS-36 and CS – 37	
CCE- Continuous and comprehensive Evaluation as follow • The continuous Comprehensive Evaluation (CCE) for each subject will be conducted by the teacher of that subject. The teacher will decide how the evaluation will be done. Usually CCE includes things like class participation, case studies and presentation, assignments, tutorials, small test (announced or surprised), quizzes and attendance or a mix of these. • Students must submit their work for internal evaluation on time to time. • Another part of CCE is the mid-term exam, which is compulsory for all students. This exam will be conducted internally by the college.	50 Marks
SEE – Semester End Examination as per the following • Practical exams may be scheduled before or after the theory examinations. Exam is conducted by Saurashtra University using external examiner (3 hours duration) • Students must prepare a practical notebook/book for the final practical examination. (The practical book serves as a record of all practical work, observations, procedures and results performed during the semester in lab. It is essential for evaluation during the final practical examination)	50 Marks

CS-38: Practical Based on CS – 35, CS-36 and CS – 37	**Total Marks - 100**	
Topics	**CCE**	**SEE**
CS-35: Mobile Application Development in Android using Kotlin	**20**	**20**
CS-36: Programming with ASP.NET	**15**	**15**
CS-37: Machine Learning with Python	**15**	**15**

CS-39: Introduction of Cloud Computing		
Unit No.	**Topic**	**Detail**
1	Introduction to Cloud Computing	• Introduction • Evolution • Benefits and berries • Cloud SPI Models • Cloud Computing vs Cluster computing • Cloud Architecture
2	Management in Cloud Computing	• Service Level Agreements (SLAs) • Quality of Service (QoS) • Scaling cloud hardware • Managing data • Cloud Security and Privacy
3	Cloud Computing Standards	• Cloud computing standards and interoperability • Technical considerations for migration to the cloud • Cloud Services • Case Studies

Reference Books:

- K Saurabh, Cloud Computing, 2nd Edition, Wiley India
- V Joysula, M Orr, G Page, (2012) Cloud Computing: Automating the VirtualizedData Center: Cisco Press.
- Mei- Ling Liu, (2004) "Distributed Computing: Principles and Application",PearsonEducation, Inc. New Delhi
- Miller M, (2008) Cloud Computing, 8th Edition, Que Publishers.
- Buyya R K, (2011) Cloud Computing: Principles and Paradigms, Wiley Press.

Course Outcomes:

- Analyses the phases of transition from classic datacenter to virtual data center and then to the cloud.
- Implement the key characteristics, services, anddeployment models of cloud.

CS-40: Internship - 100 Marks

Objectives:

- To apply theoretical concepts learned in previous semesters to practical software development.
- To gain hands-on experience in using programming languages, tools, and technologies.
- To understand and implement the complete Software Development Life Cycle (SDLC).
- To enhance problem-solving and analytical skills by designing solutions to real-life problems.
- To develop teamwork and collaboration skills through group-based project execution.
- To improve project planning and time management abilities.
- To practice documentation and reporting skills as per industry standards.
- To prepare students for major projects, internships, or professional roles in software development.

No.	Topic	Guidelines for Internship	
1	General	• Students must get prior approval from the college before starting the internship / project topic. • Choose an Internship/project topic • Projects can be individual or group-based (2 Students)(as per college policy) • Submit a project proposal with the title, objective, scope, tools/technologies to be used, and team members. • Get the proposal approved by the project guide/faculty coordinator. • Use any suitable programming language or platform • Project must be developed in any Company / Organization /Institute or computer laboratory of college. • Students must maintain a daily activity log during the internship.	
2	Documentation	Maintain a project report including the following: • Title Page • Certificate • Acknowledgment • Table of Contents • Introduction & Objective • System Analysis (Problem Definition, Feasibility Study) • System Design (DFD, ER Diagram, etc.) • Data Dictionary • Implementation (Screenshots) • Testing (Test cases and results) • Conclusion & Future Scope	
3	Submission	• Submit a soft copy (CD/ Pen drive) and hard copy of the project report. Project report must be printed on both side of the page. • Project must be submitted before two weeks of commencement of theory exam • During the project viva examination project must be run.	
4	Assessment Criteria	SEE –Exam is conducted by Saurashtra University using external examiner (examiner will evaluate the executable copy, hard copy of the project and take the viva voce)	100 Marks

BCA-5 **CS –29: Advance Java Programming(J2EE)** **Minimum following exercise should be performed by the students during the semester**	
Consider below tables for JDBC Programs: Table Name: emp Fields : empno, empnm, designation, city, salary, department Table Name: stud Fields : rollno, firstname, lastname, course, semester	
(1)	Write a program to insert a record of an employee into the emp table.
(2)	Write a program to display all the records of employees.
(3)	Write a program to display employees whose salary is greater than 50000.
(4)	Write a JDBC program to display employees who are from the city 'Rajkot'.
(5)	Write a program to display employees whose name starts with 'A'.
(6)	Write a program to display employees whose designation is manager.
(7)	Write a program to count the number of employees in the table.
(8)	Write a program to display the employee with the highest salary.
(9)	Write a program to sort employee records by empnm.
(10)	Write a program which accepts empno from the user and displays the corresponding employee record.
(11)	Write a program which accepts a department name from the user and displays the employee name along with their designation.
(12)	Write a Program which inserts a record of a student by using Prepared Statement.
(13)	Write a Program which updates a record of a student by using Prepared Statement.
(14)	Write a Program to delete a record of a student whose rollno is given by the user.
(15)	Write a program which inserts a default record of an employee using a callable statement. (Procedure without parameter)
(16)	Write a program which inserts a record of an employee using a callable statement. (Procedure with Parameter)
(17)	Write a program which displays employee’s designation by providing empno using callable statement.
(18)	Write a program which displays all the records of employees whose designation is provided by the user using a callable statement.
(19)	Write a program that performs CRUD operation on product table having fields like pid, productname, price and quatity.
(20)	Write a program to connect to a database and retrieve metadata.
(21)	Write a servlet to print Hello World.
(22)	Write a servlet that displays today's date and time.
(23)	Write a program that accepts username and after submitting the form, welcomes the user by writing “Welcome Username”.
(24)	Write a program that accepts Employee Number, Employee Name, Designation,

	Qualifications and after submitting the form display that information on a page using GET method.
(25)	Write a program that registers a student by providing rollno, name, course, semester, hobbies and after submitting the form display the information on a page using POST method. (Hint: By using getParameterNames() and getParameterValues())
(26)	Demonstrate the Servlet lifecycle with appropriate messages in each stage.
(27)	Write a servlet program that displays the Basic Header Information.
(28)	Write a program that creates a login form and after clicking on login button if login details are correct then displays welcome screen else display appropriate error message.(Hint: take username=”admin” , password=”admin”)
(29)	Write a Servlet Program that gives the following option using the radio button and perform appropriate operation (Addition, Subtraction, Multiplication, Division).
(30)	Write a program that redirects a page to google.com.
(31)	Write a program that sends a customized error message for “404 - Page Not Found”.
(32)	Write a servlet program that demonstrates the use of URL Rewriting.
(33)	Write a servlet program that demonstrates the use of Hidden Form Field.
(34)	Write a servlet program that checks if a cookie exists; if not, it creates a new cookie and adds it to the response. The servlet should then display all the cookies sent by the browser.
(35)	Write a servlet that demonstrates the use of cookies in a web application. The servlet should perform the following tasks: • Create a cookie with the name "username" and yourname as a value. • Set the cookie's maximum age to 1 day (24 hours). • Add the cookie to the response. • Retrieve any cookies sent by the client's browser and display their names and values. • If no cookies are found, display a message indicating no cookies were sent.
(36)	Write a servlet that displays a "Welcome" message if the user is visiting the site for the first time, and "Welcome back" if they have visited before. (Using Cookie)
(37)	Develop a servlet that allows the user to select a background color from a dropdown list. Save the selected color in the cookie and apply it to the response page.
(38)	Write a servlet that displays a "Welcome" message if the user is visiting the site for the first time, and "Welcome back" if they have visited before. (Using session)
(39)	Write a servlet program that accepts a user's name through a form and stores it in an HTTP session. On subsequent visits during the session, greet the user by name.
(40)	Write a Java Servlet application that authenticates users using a login form. Upon successful login, store the user’s information (like username) in the HTTP session. Display a personalized welcome page using session data. For the unsuccessful attempt display appropriate message. Also, implement a logout function that invalidates the session.
(41)	Create a servlet that counts how many times a user has visited the page during a session. Display the visit count on each load.
(42)	Develop a servlet that allows the user to select a background color from a dropdown list. Save the selected color in the session and apply it to the response page.
(43)	Write a servlet program that checks if a user has logged in. If not, redirect to a login form. Once logged in, store the username in the session and display a welcome message.
(44)	Write a JSP to print Hello World.

(45)	Write a JSP that accepts Student's details from the html page and display all the information.
(46)	Write a JSP that accepts a year from the user and print whether it is a leap year or not.
(47)	Write a JSP program to calculate simple interest by accepting values from the user.
(48)	Write a JSP page that performs a division operation. If the user enters zero as the divisor, handle the exception using a custom error page. Use the isErrorPage="true" attribute in the error page to display the exception details.
(49)	Configure an error page in web.xml that catches HTTP 404 errors (page not found). Display a user-friendly message using a JSP error page without exposing technical details.
(50)	Write a JSP application that uses <jsp:include> to include reusable pages such as header and footer into a main content page.
(51)	Create a JSP application where the user enters a favorite color; use <jsp:forward> in process.jsp to redirect to blue.jsp if the input is "blue", otherwise forward to default.jsp—ensure no output is sent before forwarding.
(52)	Create a JSP page that uses the <jsp:useBean> tag to instantiate a User JavaBean with properties name, email, and age, and display the user details on the page.
(53)	Create a JavaBean called EmployeeBean with properties for name, designation, and salary. Use the <jsp:useBean> tag to instantiate the bean with the ID employee and set its scope to session. Then, set the name, designation, and salary properties using <jsp:setProperty> tags. Finally, retrieve and display these properties (name, designation, and salary) on the page using Java expression tags (${}).
(54)	Write a JSP program to demonstrate the use of application implicit objects. On the first JSP page, use the application object to maintain a global visitor counter that increments every time the page is accessed. Display the total number of users who have visited the site since the server started.
(55)	Write a JSP program that allows the user to log in by entering their username (login.jsp). Once the form is submitted, store the username in the session object and display a personalized welcome message on a separate JSP page (welcome.jsp). The application should retain the username across multiple pages using the session and also provide an option to log out (logout.jsp), which will invalidate the session and redirect the user back to the login page (login.jsp). This program demonstrates how session tracking is used to manage user state across multiple JSP pages.
(56)	Write a JSP program that stores the user's preferred theme (e.g., "light" or "dark") using cookies. When the user selects a theme from a form and submits it, set a cookie with the selected value and redirect the user to a welcome page. On subsequent visits, retrieve the cookie and automatically apply the selected theme to the page layout.
(57)	Write a JSP that displays a "Welcome" message if the user is visiting the site for the first time, and "Welcome back" if they have visited before. (using cookie)
(58)	Write a JSP program to display current date and time using JSP Expression Language (EL).
(59)	Write a JSP application that demonstrates the use of JSP Expression Language (EL) for dynamic content rendering across multiple JSP pages. The application should begin with a form in index.jsp where the user can input their name. Upon submitting the form, the user's name should be passed to welcome.jsp using the request object. In welcome.jsp, the name passed from index.jsp should be displayed using EL.
(60)	Write a JSP program that calculates addition, subtraction, multiplication and division using expression language and display the results dynamically.
(61)	Write a JSP program to check whether it is a leap year or not using JSTL.

(62)	Write a JSP program to check whether it is an even or odd number using JSTL.
(63)	Write a JSP program to print prime numbers from 1 to 100 using JSTL.
(64)	Write a JSP program to display whether a number is divisible by 2 using JSTL.
(65)	Write a program with a Servlet that handles user input, Accepts Username as an input. Use a JSP page to welcome User by providing a username. Use the MVC pattern to separate concerns.
(66)	Write a login program. Use "admin" as the username and "admin" as the password. If the username and password are correct, display a JSP page that welcomes the user. If they are incorrect, display a JSP page that says "Incorrect username or password." Use the MVC pattern to separate concerns. (Create 2 separate views)
(67)	Write a login program. Fetch the username and password from the database. If the username and password are correct, display a JSP page that welcomes the user. If they are incorrect, display a JSP page that says "Incorrect username or password." Use the MVC pattern to separate concerns and create two separate views. (Table: Login, Fields: username, password)
(68)	Develop a Registration module using JSP, Servlet, and MVC architecture with database connectivity. Collect user details like Username, Password, and Mobile. Store the data in a database upon form submission. (Table: User, Fields : username, password, mobile)
(69)	Write a program that allows a user to submit a request to view a list of students by providing their course and semester. Use a Servlet to retrieve data from a model and display it dynamically on a JSP page. (Table: Student)
(70)	Write a Contact Management System using MVC where users can add contacts with names and phone numbers, and view all saved contacts. The Model handles contact data and interactions with the data store, the View includes a form for adding contacts and a display page for listing them, and the Controller processes input from the form, updates the model, and directs users to the appropriate view.
(71)	Write an MVC application using Servlet, JSP, and JavaBean to insert and display employee data. The application works on an emp table with fields empno, empnm, designation, and dept.
(72)	Develop a full CRUD application using the MVC model with Servlet, JSP, and JavaBean to manage student records. The application operates on studnt table containing rollno, name, course, and semester. JSP pages are used for data entry, display, update, and delete views. Servlets control the flow and interact with Beans for database transactions.
(73)	Create a simple Hibernate application that connects to a MySQL database. Define an entity class (e.g., Student with attributes rollno, name, course and semester), use Hibernate to save a new Student object, and then retrieve it using a session to display its details.
(74)	Create an entity class called Employee with attributes empno, empnm and department. Use Hibernate to insert an Employee record in the database, and then retrieve and display all employees from the database.
(75)	Create a Hibernate application with an entity class called Product that has attributes id, name, and price. Use Hibernate Query Language (HQL) to fetch all products from the database and display their id, name, and price in a list.
(76)	Create a simple login application using the Spring framework. Implement a LoginController to manage user authentication and define the necessary mappings using Spring MVC. Additionally, create templates for the login form, a success page upon successful login, and a failure page for incorrect credentials.
(77)	Develop an application using Spring Boot and Spring Data JPA to insert and display

	employee data. It manages an Employee table with empno, empnm, designation, and dept fields.
(78)	Develop an application using Spring Boot and Spring Data JPA to perform CRUD operations on a Student table without. Fields include rollno, name, course, and semester. Spring Boot simplifies the project setup, and JPA handles all database actions.

BCA-5 **CS –30: Programming in Python** **Minimum following exercise should be performed by the students during the semester**	
(1)	Write a Python program to display as following: Hello World. This is Python.
(2)	Write a Python program to work with data types (i.e., int, float, complex, boolean string types)
(3)	Write a Python program to work with following functions: id(), type(),range()
(4)	Write a Python program to work with type conversion functions.
(5)	Write a Python program to work with the following Operators in Python with suitable examples. i) Arithmetic Operators ii) Relational Operators iii) Assignment Operator iv) Logical Operators v) Bit wise Operators vi) Ternary Operator
(6)	Write Python programs to work with the following: i) input( ) ii) print( ) iii) 'sep' attribute iv) 'end' attribute v) replacement Operator ({ })
(7)	Write a Python program to work with the Conditional statements in Python with suitable examples. i) if statement ii) if else statement iii) if – elif – else statement
(8)	Write a Python program to work with the Iterative statements in Python with suitable examples. i) while loop ii) for loop
(9)	Write a Python program to work with the control transfer statements in Python with suitable examples. i) break ii) continue iii) pass
(10)	Write a Python program to work with the various ways of accessing the string. i) By using Indexing (Both Positive and Negative) ii) By using Slice Operator
(11)	Write a Python program to work with the read and write operations on a file.
(12)	Write a Python program to copy the contents of a file to another file.
(13)	Write a Python program to work with the count frequency of characters in a given file.
(14)	Write a Python program to print each line of a file in reverse order.
(15)	Write a Python program to compute the number of characters, words and lines in a file.

(16)	Write a Python program to work with the different ways of creating list objects with suitable example programs.
(17)	Write a Python program to work with the following functions/methods which operates on lists in Python with suitable examples: i) list( ) ii) len( ) iii) count( ) iv) index ( ) v) append( ) vi) insert( ) vii) extend() viii) remove( ) ix) pop( ) x) reverse( ) xi) sort( ) xii) copy( ) xiii) clear( )
(18)	Write a Python program to work with the different ways of creating tuple objects with suitable example programs.
(19)	Write Python programs to print the following Patterns: 1 22 333 4444 55555 A A B A B C A B C D A B C D E ***** **** *** ** *
(20)	Write a Python program to work with the the following functions/methods which operates on tuples in Python with suitable examples: i) len( ) ii) count( ) iii) index( ) iv) sorted( ) v) min ( )vi)max( ) vii) cmp( ) viii) reversed( )
(21)	Write a Python program to work with the different ways of creating set objects with suitable example programs.
(22)	Write a Python program to work with the following functions/methods which operates on sets in Python with suitable examples: i) add( ) ii) update( ) iii) copy( ) iv) pop( ) v) remove( )vi)discard( ) vii) clear( ) viii) union() ix) intersection( ) x) difference( )
(23)	Write a Python program to work with the different ways of creating dictionary objects with suitable example programs.
(24)	Write a Python program to work with the following functions/methods which operates on dictionary in Python with suitable examples: i) dict( ) ii) len( ) iii) clear( ) iv) get( ) v) pop( )vi)popitem( ) vii) keys( ) viii) values() ix) items( ) x) copy( ) xi) update( )
(25)	Write a Python program to return multiple values at a time using a return statement.
(26)	Write a Python program to demonstrate Local and Global variables.

(27)	Write a Python program to work with the lambda functions in Python with suitable examples.
(28)	Write a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 60 years old.
(29)	Enter the number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.
(30)	Write a python program to generate the Fibonacci series.
(31)	Write a function that reverses the user defined value using python.
(32)	Write a function to check if the input value is Armstrong or not.
(33)	Write the function for the Input number is Palindrome or not.
(33)	Write a recursive function to print the factorial for a given number
(34)	Write a function that takes a character (i.e. a string of length 1) and returns True if it is a vowel, False otherwise.
(35)	Define a function that computes the length of a given list or string.
(36)	Write a Python program to print a specified list after removing the 0th, 2nd, 3rd and 5th elements.
(37)	Write a Python script to sort (ascending and descending) a dictionary by value
(38)	Write a Python program to sum all the items in a dictionary
(39)	Implement the following Searching and Sorting techniquesin Python by using functions. i) Linear Search ii) Binary Search iii) Selection Sort iv) Bubble Sort v) Insertion vi) Merge Sort viii) Quick Sort
(40)	Write a Python program to implement encapsulation concept
(41)	Generate different plotting using PyLab. Such as Line plot, Bar chart, Pie chart, Histogram, Scatter plot.
(42)	Write a Python script to plotting a curve
(43)	Write a Python program to implement 0/1 knapsack algorithm
(44)	Write a Python program to use divide and conquer algorithm
(45)	Write a Python program to create a socket
(46)	Write a Python program to identify IP address
(47)	Write a Python program to download source code of a web page
(48)	Write a Python program to download a web page from internet
(49)	Write a Python program to download an image from internet
(50)	Create a TCP/IP Server and client

(51)	Create two-way communication such as server to client and client to server
(52)	Create a UDP server and client
(53)	Create a file server and client
(54)	Write a Python program to sending an email
(55)	Write a Python program to create GUI with use of buttons, labels, entry fields.
(56)	Write a Python program to ask confirmation or give alerts using dialogs.
(57)	Write a Python program to create a simple calculator
(58)	Write a Python program to get a list of existing databases.
(58)	Write a Python program to insert a row into a table.
(59)	Write a Python program to update a row into a table.
(60)	Write a Python program to delete a row from a table.
(61)	Create a database named dbStudent and a table named tblStudInfo: Table structure: student_id, student_name, stream, college_name, contact_number, remarks Write a Python program to insert student information.
(62)	Write a Python program to update student information.
(63)	Write a Python program to delete student information.

BCA-6 **CS –35: Mobile Application Development in Android using Kotlin** **Minimum following exercise should be performed by the students during the semester**	
(1)	Write a Kotlin program to Print "Hello World".
(2)	Write a Kotlin program to Print Name and Address (Use Println()).
(3)	Write a Kotlin Program to calculate the Area of a Circle given the radius. (Area = $\pi \times r^2$)
(4)	Write a Kotlin code to check if a year is a leap year or not. (Use if- else)
(5)	Write a Kotlin program to check if a given number is Odd or Even. (Use if- else)
(6)	Write a Kotlin program to check if a given number is Positive, Negative, or Zero. (Use if-else ladder)
(7)	Write a Kotlin Program to Print Day of the Week (1-7). (Use when)
(8)	Write a Kotlin program to print the name of the month based on a given number (1-12). (Use when)
(9)	Write a Kotlin program to take 3 numbers and find Maximum number out of that number. (Use Nested If)
(10)	Write a Kotlin program to print 5 times "Hello". (Use do – while loop)
(11)	Write a Kotlin program to print sum of 1 to 10 numbers. (Use do – while loop)
(12)	Write a Kotlin program to print odd numbers between 1 and 10. (Use for loop)
(13)	Write a Kotlin program to 10 to 1 numbers. (Use while loop)
(14)	Write a Kotlin program to print sum of all even numbers between 1 and 10. (Use while loop)
(15)	Write a Kotlin program to find the Factorial of a number using a 'while' loop.
(16)	Create an Immutable List of 3 cities and print them using a loop. (Use listOf())
(17)	Assign key to Immutable List of 3 cities and print based on key using a loop. (Use mapOf())
(18)	Create a mutable list of 3 cities and add 2 more cities after creation print using a loop. (Use mutableListOf())
(19)	Create a set of integers, add elements to it, and print all elements using a loop. (Use mutableSetOf())
(20)	Create UDF in Kotlin to perform Addition of 2 number. (Use No Argument No Return)
(21)	Create UDF in Kotlin to perform take mark of 3 subject, calculate total and percentage. (Use No Argument No Return)
(22)	Create UDF in Kotlin to take a number and print square of that number. (Use With Argument No Return)
(23)	Create UDF in Kotlin to take a number and print Cube of that number. (Use No Argument With Return)
(24)	Create UDF in Kotlin to perform Multiplication of 2 number. (Use With Argument With Return)
(25)	Create a Base class 'Person' with a properties 'name' and 'age', Derive a class 'Student' with a properties 'rollno' also add display() method in Student class which displays rollno, name and age. (Use Single Inheritance)
(26)	Create a Base class 'Person' with a properties 'name' and 'age', Intermediate base class 'Student' with a properties 'rollno', Derive a class 'Result' with a properties 'percentage' also add display() method in Result

	class which displays rollno, name, age and percentage. (Use Multilevel Inheritance)
(27)	Create a Base class 'Person' with a properties 'name' and 'age', Derive 1 a class 'Student' with a properties 'percentage' also add display1() method in Student class which displays name, age and percentage, Derive 2 a class 'Employee' with a properties 'salary' also add display2() method in Employee class which displays name, age and salary. (Use Hierarchical Inheritance)
(28)	Create two Base classes 'Mydate' and 'Mytime' having properties 'date1' and 'time1' respectively, Derive a class 'Mycalendar' which contains display() method to print date1 and time1. (Use Multiple Inheritance, Interface, Override Keyword)
(29)	Create an Abstract class 'Shape' with an abstract method area() Derive a class 'Rectangle' that implements area() method to compute the area of a rectangle. (Area = l × w)
(30)	Create a class 'Parent' with a method display(), a class 'Child' that overrides display() and calls the parent's method using 'super' keyword.
(31)	Create a class 'BankAccount' with a private property 'balance' and public methods deposit() and withdraw(). Demonstrate encapsulation.
(32)	Write a program to print "Saurashtra University" in TextView.
(33)	Write a program to print Hello World using reference to resource.
(34)	Write a program that displays your name. Use kotlin to provide your name.
(35)	Make an Android app that allows users to toast (message) "Saurashtra University" by clicking on a button.
(36)	Create an Android app that allows users to enter two strings using different EditText, click a Button to merge those two strings, and display the results in a Toast.
(37)	Create an Android app having two Buttons and one TextView to display message 'Button 1 is Clicked' or 'Button 2 is Clicked'.
(38)	Write a program for Login Application. (Take default value of username and password as admin, admin subsequently) Display appropriate message in toast if login is successful or not.
(39)	Write a program for Registration page. (Name, Gender (Radio button), Mobile Number, Address, Hobbies (Checkbox), submit button) Once submit button is clicked, display all the information in textview.
(40)	Write a program to input two numbers using different EditText, add four Buttons 'Add', 'Sub', 'Mul', 'Div' display result in TextView while clicking on a Buttton.
(41)	Write a program to input mark of 3 subjects calculates and print Total and Percentage. (Use total=m1+ m2 +m3, per=total/3)
(42)	Create an Android application to calculate and print Simple Interest. (Use si=p*r*n/100)
(43)	Write a program in which puts two Radio Button 'Male' and 'Female' inside RadioGroup. Print "Male is Radio Button" or "Female Radio Button" in Toast. (Use onCheckedChanged())
(44)	Develop an Android application with RadioButtons labeled Red, Green, and Blue. When a RadioButton is selected, the background color of the screen should change to the selected color. (Use setBackgroundColor())
(45)	Write a program that takes four Radio Button 'Image1', 'Image2', 'Image3', 'Image4' by clicking on it will set Background Image with Selected Image. (Use

	setBackgroundResource(), drawable folder)
(46)	Create an Android application with three CheckBoxes labeled 400, 700, and 900, and a Button. When the button is clicked, the app should add the values of the selected CheckBoxes and display the total in a TextView.
(47)	Write a program that takes two CheckBox 'Bold', 'Italic' and one TextView having text "Saurashtra University by clicking on CheckBox apply selected effect on TextView. (Use setTypeface())
(48)	Write a program in which add List of Cities in String.xml File, fetch it in to ListView Widget, and print selected value of ListView in Toast. (Use getResources(), getStringArray())
(49)	Create Array of Cities, add it to ListView Widget, and print selected value of Spinner in Toast. (Use ArrayAdapter())
(50)	Create an Android application that displays a ListView containing the color options: Red, Green, and Blue. When a color is selected from the list, the background color of the screen should change to the selected color. (Use setBackgroundColor(),ArrayAdapter())
(51)	Write a program that when a button is clicked, the ProgressBar will begin to progress until it reaches 100. (Use Horizontal ProgressBar, Thread)
(52)	Write a program to Add List of Cities in strings.xml file, fetch it into an AutoCompleteTextView widget, and provide suggestions (Use setAdapter())
(53)	Write a program to add List of Cities in String.xml File, Fetch it in to MultiAutoCompleteTextView Widget, and provide Multiple Suggestions. (Use setAdapter(),setTokenizer())
(54)	Write a program that uses a DatePickerDialog in an Android app to allow the user to select a date. Once a date is chosen, display the selected date in a TextView using Kotlin.
(55)	Write a program that uses a TimePickerDialog in an Android app to allow the user to select a time. Once a time is chosen, display the selected time in a TextView using Kotlin.
(56)	Write a program that displays AlertDialog() having Ok button only, when the user clicks on it will update the TextView with "Ok button is Clicked". (Use AlertDialog.BUTTON_NEUTRAL)
(57)	Write a program that displays AlertDialog() having Yes and No buttons, when the user clicks on it will update the TextView with "Yes button is Clicked" or "No button is Clicked". (Use setPositiveButton, setNegativeButton)
(58)	Write a program that displays AlertDialog() having Ok and Cancel buttons, when the user clicks on it will update the TextView with "Ok button is Clicked" or "Cancel button is Clicked". (Use setPositiveButton, setNegativeButton)
(59)	Write a program that creates two activities: First Activity – It has an EditText to enter a name and a Button. When the button is clicked, it opens the second activity. Second Activity – It has a TextView that shows the name received from the First Activity. (Use Intent)
(60)	Develop an android app to pass data to another activity. (Use Intent)
(61)	Write a program that creates two activities: First Activity – It has an EditText to enter a name and a Button. When the button is clicked, it opens the second activity. Second Activity – It has two TextView 1st shows Hello and 2nd shows the name

	received from the First Activity. (Use Intent)
(62)	Create an Android application that performs a Horizontal translation animation on an ImageView, where the image moves from left to right across the screen. (Use anim folder, AnimationUtils, loadAnimation)
(63)	Create an Android application that performs a Vertical translation animation on an ImageView, where the image moves from up to down across the screen. (Use anim folder, AnimationUtils, loadAnimation)
(64)	Make an Android application that performs animation on ImageView to rotate image in a Clockwise. (Use anim folder, AnimationUtils, loadAnimation)
(65)	Make an Android application that performs animation on ImageView to rotate image in an Anti-clockwise. (Use anim folder, AnimationUtils, loadAnimation)
(66)	Create an Android application that applies a scale animation to an ImageView, creating a zoom-in effect on the image when triggered. (Use anim folder, AnimationUtils, loadAnimation)
(67)	Create an Android application that applies a scale animation to an ImageView, creating a zoom-out effect on the image when triggered. (Use anim folder, AnimationUtils, loadAnimation)
(68)	Create Android application to apply Frame By Frame Animation.
(69)	Write a program in which set 3 columns GridView with 6 cities name, clicking on it will display selected city name in Toast. (Use ArrayAdapter, onItemClick())
(70)	Write a program in which set 2 columns GridView with 4 color name, clicking on it will apply selected color in Background. (Use ArrayAdapter, onItemClick())
(71)	Write a program in which divide MainAcitivity in two Fragment. 1st Fragment prints “Saurashtra” and 2nd Fragment prints “University” in TextView. (Use Fragment())
(72)	Write a program in which divide MainAcitivity in three Fragment. 1st Fragment prints “Saurashtra”, 2nd Fragment prints “University” and 3rd Fragment prints “Rajkot” in TextView. (Use Fragment())
(73)	Create an Android application to demonstrate Activity Life Cycle. (Use onCreate(), onStart(), onResume(), onPause(), onStop(), onDestroy() etc.)
(74)	Make Android application to share simple text using concept of Shared Intent. (Use setAction(), setType(), createChooser())
(75)	Write a program in which add name, city and gender in EditText, clicking on Save button will create and save data in key-value format. (Use SharedPreference())
(76)	Write a program in which adds username and email in EditText, clicking on Save button will create and save data in key-value format. (Use SharedPreference())
(77)	Create an Android application that allows reading from and writing data to internal storage.
(78)	Develop an Android app that enables users to save data to the device's external storage and retrieve it later.
(79)	Create an Android application that allows the user to insert student records (roll no, name, course) into a local SQLite database, and display the stored records in a ListView.
(80)	Develop an Android application that provides search functionality, allowing the user to search for student records by roll no, and display the matching results from the database.

(81)	Write a program that performs CRUD operation on Student database. (Use SQLite database)
(82)	Create an android application to perform create, insert, and delete operation on Employee database having fields like eid, enm, phno, sal (Use SQLite database)
(83)	Write a program that performs CRUD operation on Department table having fields like deptid, deptnm.
(84)	Create an android application to perform create, insert, and delete operation on Book Master database having fields like book_id, book_name, author_name, book_price (Use SQLite database)
(85)	Create an Android application to perform Login Activity having EditText (username and password), buttons (Login, Save). When user clicks on "Login" button will check username and password in database if it is matched with inputted value then switch to next Activity. When user clicks on "Save" button will insert record in Login table. (Use SQLite database)
(86)	Create an Application to check network is available or not. (Set Permission ACCESS_NETWORK_STATE, INTERNET, ConnectivityManager)
(87)	Create an Android application that provides your cell details like (IMEI, SIM Number, and Operator Name etc.). (Set Permission READ_PHONE_STATE, TelephonyManager)
(88)	Create an Android application to Listen phone state and check phone is Idle, Off-hook or Ringing. (Use StateMonitor, PhoneStateListener())
(89)	Create an Android application that opens the provided URL in WebView control.
(90)	Create an Android application that displays a webpage in a WebView component and shows a ProgressBar while the page is loading.
(91)	Create an Android application that sends sms to the provided mobile numbers.
(92)	Create an Android application to notify user with status bar. (Use NotificationManager)
(93)	Create an Android application to update notification of user in status bar. (Use Notification, Builder, notify())
(94)	Create an Android application that notifies the user with status bar and vibration in device. (Set Permission VIBRATE)
(95)	Create an Android activity that includes a ToggleButton. When the button is toggled: Display a Toast message saying "Service is Started" when it is turned ON. Display "Service is Destroyed" when it is turned OFF. (Use Service, onStartCommand(),onDestroy())
(96)	Create an Android application to fetch current latitude and longitude of device. (Set Permission ACCESS_COARSE_LOCATION, ACCESS_FINE_LOCATION, INTERNET)
(97)	Display Country name, City name and Postal Code from latitude and longitude. (Use Geocoder, getFromLocation())
(98)	Create an Android application that fetches student data from a MySQL database using a PHP script. The PHP script queries the database and returns the data in JSON format, which is then parsed and displayed in the Android app using a ListView.

BCA-6 **CS –36: Programming with ASP.NET** **Minimum following exercise should be performed by the students during the semester**	
(1)	WAP (1) to display "Hello World" on web page (2) Demonstrate client-side validation using a RequiredFieldValidator
(2)	Create a form with Textbox, Button, Checkbox and retrieve values and demonstrate file upload functionality using the FileUpload Control
(3)	WAP to use RegularExpressionValidator to validate phone number format.
(4)	WAP which (1) implements various validation controls. Add a ValidationSummary to display error messages (2) compare password and confirm password. (Use CompareValidator)
(5)	WAP to (1) create a web form with ListBox and display selected items on button click (2) Demonstrate RangeValidator to accept date between 01/01/2000 to Current Date.
(6)	Create web form with Textbox, Label and Button Control. (1) Display TextBox content in a Label when the Button is Clicked (2) Display TextBox content in proper case in the Label when the Button is double Clicked
(7)	Develop a Student Login Form with use of RequiredFieldValidator, CompareValidator Controls.
(8)	Develop a form with a Panel Control that shows / hides based on CheckBox.
(9)	Implement an Interactive Registration Form with Validations.
(10)	Use RegularExpressionValidator for email, phone and zipcode validation.
(11)	Write a program to implement two cascading DropDownLists in ASP.NET Web Forms. When a country is selected from the first DropDownList, the second DropDownList should display states related to that country.The list of countries and their corresponding states should be statically populated.
(12)	Write a program to create two ListBoxes. The first ListBox contains available courses. Allow the user to select courses and move them to the second ListBox using ">>" and "<<" buttons.
(13)	Write a program that allows user to enter a new city name in TextBox. When the user clicks "Add" the new city should be added to the DropDownList dynamically.
(14)	Write a program to group three RadioButtons (LENOVO, SONY, PHILIPS) and display the selected name in a Label when a button is clicked.
(15)	Write a program with three RadioButtons (India, US, NewZealand). When user selects one and clicks a button, display related image (or flag) using the Image control.
(16)	Write a program with a TextBox for password input and a CheckBox labelled "Show Password". When checked, change the TextMode of the TextBox from Password to SingleLine.
(17)	WAP to (1) store a message in ViewState and display it (2) store and retrieve any greeting message using Cookies.
(18)	WAP to (1) implement session-based login functionality (2) to implement Session Timeout and display message when expired.
(19)	Demonstrate (1) QueryString to pass name from one page to another (2) the use of hidden fields for data storage
(20)	WAP to (1) retain values on postback using ViewState (2) Create a page to count user

	visits using Application State.
(21)	WAP to (1) save user login status in Session (2) demonstrates how ViewState automatically preserves the value of TextBox across postbacks without any explicit coding for state management.
(22)	Build a shopping cart using Session to store cart items.
(23)	Create a webform which uses QueryString and Session together.
(24)	WAP to demonstrate the use of ViewState to store user preferences (like color / font)
(25)	Create a multi-page application that demonstrates the use of QueryString, Session and Application State.
(26)	Write a program which has a DropDownList for color selection, a Button to set the cookie and a Label to show the retrieved preference.
(27)	Create a webpage where users can select their preferred website theme (“Light Mode” or “Dark Mode”) from a dropdown list. (Use Cookie). When the user revisits the page, show only last selected theme
(28)	Create WebForms (1) Collect Basic Details like name, email, contact (2) Collect Educational Details like HSC Board Name, HSC Passing Month-Year, HSC Percentage (3) Display all collected data from previous pages for review. (Use Session State)
(29)	Create a countdown timer using Session storage.
(30)	Create Login Form. Develop “Remember Me” feature for username using cookies.
(31)	Create a login page that uses Session to track logged-in users.
(32)	Create a connection to a database and display connection status. And also WAP to retrieve and display records using SqlDataReader
(33)	WAP to use GridView to display records from database table and also WAP to populate Country DropDownList and based on country selection – State DropDownList. (Country and State names must be from database tables only)
(34)	WAP to populate ListBox with product names from database and display selected product name to Label.
(35)	WAP to demonstrate using Repeater control to display formatted database data.
(36)	WAP to demonstrate use DataSet to fill data and display in GridView also implement paging.
(37)	WAP to implement login system with user authentication using a database.
(38)	Create a WebForm with Search TextBox. Find out complete record from database according to entered details in Search TextBox.
(39)	Create an ASP .NET application that performs CRUD operations.
(40)	Create an ASP .NET application to display records / data in GridView. (Use Disconnected Architecture).
(41)	Create an ASP .NET application which uses SqlDataSource with parameterized queries.
(42)	Develop a Web Form that shows related data into two GridViews (Master – Detail)
(43)	Develop a Web Form that uses DataView to filter records.
(44)	Create an ASP .NET application that display product data in Repeater with custom template.
(45)	Create an ASP .NET application to register student detail and store to database. (Use Validation wherever required).
(46)	Create an ASP .NET application to retrieve data based on search text. (Use StoredProcedure to get data from Database).

(47)	WAP to display all products in a GridView (Disconnected). Allow the user to select a product from the DataGridView and update its StockQuantity via a separate input field.
(48)	Create a web application in ASP.NET that displays data in a GridView using a connected architecture. Implement sorting and paging functionality for the displayed data.
(49)	Write an ASP .NET application to develop a Master Page. Add Header and Footer to it also add it to multiple content pages.
(50)	Write an ASP .NET application to create a theme with a stylesheet and apply it to webpage using the <Themes> element in web.config.
(51)	Create an ASP .NET page to enable Page Output Caching for 30 seconds and also demonstrate the use of @OutputCache
(52)	Write code for an ASP .NET application to create a theme with different color styles and apply it.
(53)	Create an ASP .NET page to demonstrate Data Caching.
(54)	Develop an ASP .NET Web Form that utilizes full page output caching.
(55)	Develop an ASP.NET Web Form that stores and retrieves a simple string using the HttpContext.Current.Cache object.
(56)	Develop an ASP .NET Web Form that demonstrates explicit removal of an item from the HttpContext.Current.Cache.
(57)	Create an apply two distinct ASP .NET Themes to a web application, configuring one as a default. (can use LinkButtons like “Apply Aqua Theme” and “Apply Forest Theme”)
(58)	Create an ASP .NET application to apply themes. Write skin file code for styling of ASP .NET controls.
(59)	Develop a WebUserControl with output caching and integrate it into a web form to demonstrate fragment caching.
(60)	Develop an ASP .NET Web Form that stores and retrieves a collection of custom objects in HttpContext.Current.Cache with an absolute expiration policy.
(61)	Develop an ASP .NET Web Form that stores and retrieves a collection of custom objects in HttpContext.Current.Cache with a sliding expiration policy.
(62)	Create ASP .NET application which apples Themes with Skin File.
(63)	Create small ASP .NET application to store, search and retrieve student details. (Use Master Page and Themes concept)
(64)	Create a simple login / logout system that integrates with application’s theming.
(65)	Develop a WebUserControl that receives different default styles based on a theme’s skin file.
(66)	Develop an ASP .NET Web Application where a user’s theme change is confirmed via JavaScript before applying it permanently.
(67)	Write an ASP .NET application to (1) read and display XML data (2) writes dataset contents into an XML file.
(68)	Write an ASP .NET Web Form that generates a simple XML file and reads a single configuration value from an XML file and display it.
(69)	Write an ASP .NET code to develop asmx web service that provides a method with text “Hello World”.
(70)	Write an ASP .NET Web Form that displays an application name configured in web.config.

(71)	Write an ASP.NET application which (1) consumes a simple Web Service and display output (2) shows error handling using Web. Config.
(72)	Develop an ASP .NET Web application that (1) reads from connectionStrings and a custom web.config section. (2) reads AppSettings from Web.config and display it in Label.
(73)	Develop a Web Service that returns product details and consume it.
(74)	Write an ASP .NET application which fills DropDownList dynamically. (Use XML Data Only)
(75)	Write an ASP .NET Web Form that demonstrates page-level tracing.
(76)	Develop an ASP .NET Web Application which redirects to a custom error page if any exception occurs.
(77)	Develop an ASP .NET Web Login Form (Use Web.Config – Authentication and Authorization)
(78)	Develop an ASP .NET Web Application for managing list of Employees using XML as the data store. (Consider employee.xml with id, name, designation, salary)
(79)	Write an ASP .NET code which reads and updates XML data dynamically in an ASP .NET application.
(80)	Develop an ASP .NET Web Application with custom error pages for 404, 500 and display basic error info.
(81)	Develop an ASP .NET Web Application with a login, secure user area and a restricted admin area.
(82)	Write an ASP .NET code which uses XmlDataSource to display XML data in a TreeView control.

BCA-6 **CS –37: Machine Learning with Python** **Minimum following exercise should be performed by the students during the semester**	
(1)	Write a Python program/script to make a Pandas DataFrame with two-dimensional list
(2)	Write a Python program to Create a pandas column using for loop
(3)	Write a Python program toChange column names and row indexes in Pandas DataFrame
(4)	Write a Python program to Load different kind of datasets using scikit-learn library
(5)	Write a Python program to Extract the specified rows and columns from the dataset using Pandas
(6)	Write a Python program toHandle missing values using Imputer class with mean strategy
(7)	Write a Python program to Encode categorial data using label encodingtechnique
(8)	Write a Python program toEncode categorial data using one hot encodingtechnique
(9)	Write a Python program to splitting dataset into Training set and Test set
(10)	Write a Python program toPerform feature scaling using standardizationtechnique
(11)	Write a Python program toPerform feature scaling using normalizationtechnique
(12)	Write a Python program toCreate a matrix using numpy and work around
(13)	Write a Python program toPerform mean removal using preprocessing techniques
(14)	Write a Python program toPerform scaling and generate datapoints in a range
(15)	Write a Python program toCreate a vector using binarization technique
(16)	Write a Python program toPerform linear regression using different relationships
(17)	Write a Python program toEvaluate linear regression model using different metrics
(18)	Write a Python program on linear regression model using advertising sales channel data
(19)	Write a Python program toPerform data cleaning processes such as identify null values and outliers
(20)	Write a Python program toGenerate some visualizations to get the detailed insights
(21)	Write a Python program toWorking with heatmap to understand correlation concepts in Machine learning
(22)	Write a Python program toPerforming a summary operation
(23)	Write a Python program toBuilding simple classifier using anyone dataset
(24)	Write a Python program toPerform standard normal distribution using simple classifier
(25)	Write a Python program toBuilding alogistic regression model with use of diabetes datasets
(26)	Write a Python program toEvaluate logistics regression model using accuracy metrics
(27)	Write a Python program toEvaluate a regression model using confusion matrix
(28)	Write a Python program toBuilding a model using Naïve bayes classifier
(29)	Write a Python program toVisualize the training setand test set result (use normalization technique)
(30)	Write a Python program toPredict if cancer is Benign or malignant using SVM algorithm
(31)	Write a Python program toBuild a model using K-means algorithm
(32)	Write a Python program toFind the optimum number of clusters using elbow technique
(33)	Write a Python program toPlot the cluster center using different data points
(34)	Write a Python program toImplement Mean shift clustering algorithm to work with non-parametric clustering
(35)	Write a Python program toUse bandwidth and bin seeding concept to improve mean shift

	clustering algorithm
(36)	Write a Python program toBuild a model with use of agglomerative clustering
(37)	Write a Python program toCreate a linkage matrix using agglomerative clustering algorithm
(38)	Write a Python program toImplement NLTK library and download relevant data
(39)	Write a Python program toImplement stemming concept with using PorterStemmer
(40)	Write a Python program toImplement lemmatization technique to extract the base form of words
(41)	Write a Python program toCreate a chunk parser
(42)	Write a Python program toImplement the structure of sentence
(43)	Write a Python program toEvaluate the grammar using parser
(44)	Write a Python program toGenerate a grammar tree with use of sentence
(45)	Write a Python program toImplement computer vision using OpenCV
(46)	Write a Python program toWork around computer vision relevant python libraries
(47)	Write a Python program toUse of imread(), imshow(), and imwrite()
(48)	Write a Python program toDetect faces from an image using haar-cascade classifier
(49)	Write a Python program toDetecting different objects from a face such as face, eyes
(50)	Write a Python program toDetect a face from a recorded video
(51)	Write a Python program toDetect a face using live streaming