Structure & Syllabus for Semester-VI

BACHELOR OF COMPUTER APPLICATIONS (BCA) Programme

Gujarat University

2025 - 2026

As per NEP 2020 CURRICULUM AND CREDIT FRAMEWORK FOR UNDERGRADUATE PROGRAMMES, UGC

&

Resolution No. KCG/admin/2023-24/0607/kh.1

of

Education Department, Govt. of Gujarat

STRUCTURE FOR SEMESTER - 6
GUJARAT UNIVERSITY
Bachelor of Computer Applications (B.C.A.)
(as per NEP 2020)

COURSE:	**Bachelor Of Computer Applications (B.C.A.)**
MAJOR:	**COMPUTER APPLICATIONS**
MINOR:	**1. COGNITIVE MATHS**
	2. DATA SECURITY
	3. WEB TECHNOLOGIES

SEMESTER – 6

MAJOR

CODE	COURSE	CREDITS
DSC-C-BCA-361T	Computer Netwwork	4
DSC-C-BCA-362T	Cloud Computing	4
DSC-C-BCA-363P	System Development Project (External Evaluation of SDP1 & SDP2)	4

MINOR

CODE	COURSE	CREDITS
DSC-M-BCA-364P	Advanced Python Programming	4
	OR	
DSC-M-BCA-364P	Digital Forensic Practical	4
	OR	
DSC-M-BCA-364P	Web Applications Development Framework using Laravel	4

ABILITY ENHANCEMENT COURSE ()

AEC-BCA-365	(Any ONE Course to be selected from the Basket)	2

SKILL ENHANCEMENT COURSE
(Any ONE Course. Course can also be chosen from Basket)

SEC-BCA-366	Internship in Major Specific Course	4
	TOTAL CREDITS	**22**

Semester – VI

BACHELOR OF COMPUTER APPLICATIONS

MINORS

1. COGNITIVE MATHEMATICS (CG)
2. DATA SECURITY (DS)
3. WEB TECHNOLOGIES (WTECH)

Course Name: Computer Network

Course Code: DSC-C-BCA-361 T

Credits: 4

Course Outcomes:
The aim of this course is to enable students to

- To become familiar with the fundamentals of data communication and networking.
- To understand different network technologies.
- To get insights into different advanced network technologies that can be used to connect different networks.
- To understand functions of different layer

Prerequisites:

- NIL

Contents:

Unit	Particulars	Hours	Credits
1	**Introduction to Computer Networks:**		
	Introduction to data communication, Computer Networks, Characteristics, Components, Data Representation **Types of Networks:** PAN, LAN, CAN, MAN, WAN **Topologies:** Bus, Ring, Star, Mesh, Hybrid, Tree **Key terms:** Message, Sender, Receiver, Medium, Jitter, Types of data **Network Model:** OSI Model, TCP/IP Model, Comparison of OSI and TCP/IP Models.	15	1
2	**Physical Layer**		
	Data and Signals, Analog Signals and Digital Signals **Key Terms**: Period, Frequency, Phase, Bandwidth, Bit rate **Transmission Impairment**: Attenuation, Distortion, Noise **Performance** (Throughput, Latency, Transmission Time, Queuing Time **Multiplexing**: Introduction, Types of multiplexing - FDM, TDM, WDM **Transmission Media**: **Guided Media**: Twisted pair Wire, Coaxial Cable, Optical Fibers **Unguided Media**: Introduction	15	1

	Switching: Introduction, Circuit Switched Networks, Packet Switching (Datagram Networks, Virtual Circuit Network), Message switching **Introduction to Networking Devices:** Repeater, Amplifier, HUB, Switches, Bridges, Routers		
3	**Data-link layer**		
	The Logical Link Layer Introduction, **Error**: Types of Error**, Error Detection** : Checksum , Parity Check, CRC, Hamming code **Framing:** Character Count, Byte Stuffing, Bit Stuffing, Physical Encoding **Data Link Layer Protocols:** UTOPIAN simplex protocol, Stop-and-Wait for error-free and Noisy channel, Sliding Window, Go-Back N Protocols. **The MAC Layer** **Medium access Protocols:** ALOHA (Pure and Slotted), CSMA, CSMA-CD, CSMA-CA **Controlled Access Protocols:** Reservation, Polling, Token Passing **Channelization Protocols**: FDMA, TDMA, CDMA.	15	1
4	**Other layers**		
	Network Layer **Routing Algorithms:** Introduction, Shortest Path Problem, Flooding **IP Addressing:** Introduction, IPv4, Class A, B and C addressing, Introduction to IPv6. **Transport Layer:** Introduction**,** TCP and UDP, TCP Connection establishment and release. **Application Layer**: Introduction, DNS, DHCP, HTTP, SMTP, POP3 and IMAP.	15	1

Textbook:

1. Data communications and networking – 5th Edition
   Publisher: McGraw Hill
   By Behrouz A. Forouzan

References:

1. Computer networks – 5th Edition
   Publisher: Pearson
   By Andrew S. Tanenbaum

2. Business data communication
   Publisher: Cengage publications
   By Selly Cashman

3. Data Communications and Networks, 2nd Edition
   Publisher: McGraw Hill
   By Achyut S Godbole, Atul Kahate

Other Web References:

https://www.geeksforgeeks.org/data-communication-definition-components-types-channels/
https://www.geeksforgeeks.org/framing-in-data-link-layer/
https://www.tutorialspoint.com/data_communication_computer_network/aloha_protocol.htm
https://www.geeksforgeeks.org/tcp-and-udp-in-transport-layer/
https://www.geeksforgeeks.org/protocols-application-layer/

Accomplishments of the student after completing the Course:

After completion of this course Student would be able to

- Understand the fundamentals of data communication and networking.
- Understand types of networks, their functions and topologies
- Understand use of different layers for data communication
- Understand advanced network technologies.

Course Name: Cloud Computing

Course Code: DSC-C-BCA-362 T

Credits: 4

Course Outcomes:

The aim of this course is to enable students to

- uncover the core concepts of cloud computing
- provide ample foundations to enable further study, research and implementation of cloud computing services.

Prerequisites:

- NIL

Contents:

Unit	Particulars	Hours	Credits
1	**Introduction to Cloud Computing** **Cloud Computing at a glance:** The vision of cloud computing, Defining a cloud, A closer look, The cloud computing reference model, Characteristics and benefits, Challenges ahead **Historical Developments:** Distributed systems, Virtualization, Web2.0 **Build Cloud Computing Environments:** Computing platforms and technologies - Amazon Web Services (AWS), Google AppEngine, Microsoft Azure, Hadoop	15	1
2	**Principles of Parallel and Distributed Computing** **Eras of computing** **Parallel vs. Distributed Computing** **Elements of parallel computing:** What is parallel processing?, Hardware architectures for parallel processing, Approaches to parallel programming **Elements of distributed computing:** General concepts and definitions, Components of a distributed system, Architectural styles for distributed computing - System architectural styles, Models for inter process communication **Technologies for distributed computing:** Service-oriented computing	15	1
3	**Virtualization** **Introduction** **Characteristics of virtualized environments** **Taxonomy of virtualization techniques:** Execution virtualization - Machine reference model, Other types of virtualization	15	1

	Virtualization and cloud computing **Pros and cons of virtualization** **Technology examples:** VMware (full virtualization) – Virtualization Solution - End-user (desktop) virtualization		
4	**Cloud Computing Architecture** **Introduction** **The cloud reference model:** Architecture, Infrastructure and hardware-as-a-service. Platform-as-a-service, Software-as-a-service **Types of clouds:** Public clouds, Private clouds. Hybrid clouds, Community clouds **Economics of the cloud**	15	1

Textbook:

2. Mastering Cloud Computing: Foundations and Applications Programming
   By Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi
   Publication: McGraw Hill Education, 2013

References:

4. Essentials of Cloud Computing
   By K. Chandrasekaran
   Publication: CRC press, 2015

Other Web References:

Supplemental materials for instructors or students can be downloaded from elsewere:

1. http://store.elsevier.com/product.jsp?isbn=9780124114548
2. https://www.tutorialspoint.com/cloud_computing/index.htm
3. https://www.geeksforgeeks.org/cloud-computing/
4. https://www.w3schools.in/cloud-computing/cloud-computing/

Accomplishments of the student after completing the Course:

After completion of this course Student would be able to

- Visualize the various cloud computing environments
- Understand the core concepts of parallel, distributed computing and its architecture.
- Learn and implement virtualization environments
- Articulate the insights of cloud computing architecture and its types

---

Course Name: System Development Project

Course Code: DSC-C-BCA-363 P

Credits: 4

System Development Project as a partial fulfillment in B.S.(BCA) course involves creating a functional software system, encompassing stages like system analysis, design, coding, testing, and implementation

Objective

- A system development project is a practical, hands-on experience where B.S (BCA) students apply their programming and software development knowledge to build a complete software system.
- It's a core component of the BCA curriculum, allowing students to demonstrate their skills and gain real-world experience.
- The project typically involves a team of students working under the guidance of a faculty mentor.
- This course is designed to provide the students an experience of working with a client organization from the initial request through final design and development of prototype software.

Timeline

- Define project scope, assess feasibility, and establish a project schedule.
- Requirements Analysis: Understanding the needs and functionalities of the proposed system.
- System Design: Planning the structure, architecture, and components of the system.
- Preparing System Flow Diagram, Entity Relationship Diagram, Data Flow Diagram / UML Diagram, Building Data Dictionary
- Coding/Implementation: Writing the actual software code using programming languages like Java, Python, or C++.
- Testing: Thoroughly testing the system to identify and fix bugs and errors.
- Deployment: Making the software system available for use.
- Documentation: Creating documentation for the system, including user manuals and technical specifications.

Aim

1. Get exposure to software development process by choosing a typical business/scientific/administrative/system application.
2. Get some experience in working with a client organization.
3. Gain experience in working in a group to successfully develop the deliverables.

Guidelines

- Each group should have a minimum of 2 and maximum 3 students
- One day off in a week, to work on the project
- At least three hours must be allotted in the weekly timetable for discussion with faculty (college guide)
- The location of the organization should be restricted to Ahmedabad City
- The students can do project related work either in the organization, college or Home

Detailed Course Contents and Guideline for students:

1. Developing System Design

- **Students are expected to work on the following during the semester.**
  - Doing System Analysis and Design
  - Preparing System Flow Diagram (if applicable)
  - Developing Entity Relationship Diagram (For Relational Model)
  - Developing Data Flow Diagram / UML Diagram
  - Building Data Dictionary (For relational model)
  - Writing code for the project
  - Implementation and testing of the code

- **Where to look for a Project?**
  - Government Organizations
  - Local Self Government (Municipalities, Panchayats, Urban Development Authorities etc.) or public / private bodies or NGOs.
  - Public Sector Organizations
  - Educational institutes
  - Trading/Business houses
  - Private Organizations
  - Software Consultancy companies (only if the project work seems to be original and beneficial)
  - A challenging in-house software project.

- **Which Projects to Avoid?**
  - The project of system study
  - Involves only modification in existing software, such as porting of software or few updates
  - Involves only data storage and retrieval without any processing.

- **Preferred Projects:**
  - Will be such as that caters to Innovative areas/ideas
  - Use of emerging technologies
  - Challenging uses of Communication and the Internet
  - Scientific applications
  - Graphics applications
  - Systems software and utilities
  - Embedded software
  - ERP modules
  - Web development projects
  - Mobile Application Development
  - Database Management Project
  - Artificial Intelligence & Machine Learning based projects
  - Game Development
  - Data Science Analytics
  - IOT Based Projects etc

- **Preferred Tools:**
  - Students should feel free to use the tools of their choice subject to permission of the organization.
  - Working on any acceptable project would give good exposure to use of analytical tools, programming skills and development tools. Hence, any programming or development environment should be acceptable.

➢ **Deliverables by the students:**

- At the end of the semester, the student should be able to work on the identified project and submit the documentation (hard copy) and the presentation of the system project details
- **Documentation:**
  - A hard copy of the documentation should consist of the following:
  - Cover Page
  - Company Certificate
  - College Certificate
  - Acknowledgement
  - Screen layouts
  - Report layouts
  - Sample coding
  - Future Enhancements (optional)
  - Conclusion
  - Bibliography
  - A log-sheet of reporting to the college faculty must be maintained.
  - Index (with page nos.)
  - Organization / Company Profile
  - Project Profile
    - Existing System
    - Proposed System
    - Development Tools and Technology used
  - System Flow Diagram (if applicable)
  - UML Diagram/Data Flow Diagram
  - Entity Relationship Diagram
  - Data Dictionary/Table Design
- Live Demo of the Project must be shown at the time of presentation, though not compulsory, students can be evaluated accordingly by the examiners.

➢ **Presentation:**

Presentations can be prepared through slides using any Open-Source PowerPoint/Prezi/Canva or any other multimedia tool, covering the work shown in the documentation.

Presentation is a must, but if students are showing the demo of the project, then faculty might not go through the presentation if there is a time constraint.

Coding can be checked by the external faculty, to check for genuineness of the work.

During viva examination, students will be expected to satisfactorily answer questions pertaining to the project profile, diagrams, data dictionary, the tools used, the process, the reports /forms created, and the results achieved.

Checking the payment part is not compulsory

Students will be evaluated based on the questions answered, documentation, coding standards, clarity of concepts, presentation skills and group coordination.

Structure & Syllabus for Semester-VI

BACHELOR OF COMPUTER APPLICATIONS (BCA) Programme

Gujarat University

2025 - 2026

As per NEP 2020 CURRICULUM AND CREDIT FRAMEWORK FOR UNDERGRADUATE PROGRAMMES, UGC

&

Resolution No. KCG/admin/2023-24/0607/kh.1

of

Education Department, Govt. of Gujarat

STRUCTURE FOR SEMESTER - 6
GUJARAT UNIVERSITY
Bachelor of Computer Applications (B.C.A.)
(as per NEP 2020)

COURSE:	**Bachelor Of Computer Applications (B.C.A.)**
MAJOR:	**COMPUTER APPLICATIONS**
MINOR:	**1. COGNITIVE MATHS**
	2. DATA SECURITY
	3. WEB TECHNOLOGIES

SEMESTER – 6

MAJOR

CODE	COURSE	CREDITS
DSC-C-BCA-361T	Computer Netwwork	4
DSC-C-BCA-362T	Cloud Computing	4
DSC-C-BCA-363P	System Development Project (External Evaluation of SDP1 & SDP2)	4

MINOR

CODE	COURSE	CREDITS
DSC-M-BCA-364P	Advanced Python Programming	4
	OR	
DSC-M-BCA-364P	Digital Forensic Practical	4
	OR	
DSC-M-BCA-364P	Web Applications Development Framework using Laravel	4

ABILITY ENHANCEMENT COURSE ()

AEC-BCA-365	(Any ONE Course to be selected from the Basket)	2

SKILL ENHANCEMENT COURSE
(Any ONE Course. Course can also be chosen from Basket)

SEC-BCA-366	Internship in Major Specific Course	4
	TOTAL CREDITS	**22**

Semester – VI

BACHELOR OF COMPUTER APPLICATIONS

MINORS

1. COGNITIVE MATHEMATICS (CG)
2. DATA SECURITY (DS)
3. WEB TECHNOLOGIES (WTECH)

Course Name: Web Application Development Framework Using Laravel

Course Code:

Credits: 4

Course Outcomes:

On the completion of the course students will:

- To understand the LARAVEL framework
- To understand how to implement LARAVEL
- To understand how to design and develop responsive web application using LARAVEL

Prerequisites:

HTML, CSS and Basics of PHP

UNIT	TOPIC/SUBTOPIC	TEACHING HOURS	Credits
1	**Introduction to LARAVEL**	15	01
	Why Use a Framework? A Short History of Web and PHP Frameworks, What's So Special About Laravel?, How It Works, Why Laravel? Setting Up a Laravel Development Environment: System Requirements, Composer, Local Development Environments, Creating a New Laravel Project, Laravel's Directory Structure, Configuration, An Introduction to Artisan Tool, Basic Artisan commands.		
2	**LARAVEL Basics**	15	01
	Router and Controllers: Route Definitions, Route Groups, Views, Controllers, Route Model Binding, Route Caching, Form Method Spoofing, CSRF Protection, Redirects, Aborting the Request, Custom Responses **Blade Template:** Echoing Data, Control Structures (Conditionals & Loops), Template Inheritance: @include, @extends, @section, @yield, @stack and @push, View Composers and Service Injection, Custom Blade Directives.		
3	**Collecting and Validating User Data**	15	01
	Laravel Migration Laravel migration, migration structure, generating migrations, migration commands **Laravel file upload, laravel session** **Laravel database** Laravel database, laravel eloquent, laravel relationship model **Eloquent orm models:** Naming conventions, table name & primary keys, timestamps **Basic operations:** create, retrieve, update, delete using models, displaying data from models in views		
4	**User Authentication and Authorization**	15	01

	User authentication and authorization : Using the auth() global helper and the auth façade **Validation** Defining the routes, creating the controller, writing the validation logic, displaying the validation errors, array validations, creating new validators, error messages & custom errors **Available validators:** Accepted, after (date), alpha, alpha dash, alpha numeric, array, before (date), between, boolean, date, date format, different, digits, digits between, e-mail, exists(database), image (file), integer, max, min, not in, numeric, regular expression, required, string custom validation rules.		

Laravel Project/Practical: To-Do List Web Application

Project Objective:
Design and develop a To-Do List Web Application using the Laravel framework, starting from scratch and implementing features such as adding, displaying, editing, validating, and deleting tasks.

Phase	Project Tasks/Practical Questions
1	**Project Setup & Routing Basics (Tasks 1–10)**
	1. Install Composer on your system and verify using terminal. 2. Create a new Laravel project named `todoapp` using Composer. 3. Run the Laravel project using `php artisan serve`. 4. Explore the default folder structure and understand the MVC architecture. 5. Create a route `/welcome` that returns a simple welcome message. 6. Create a route `/tasks` that returns a hardcoded array of tasks. 7. Create a route `/task/{name}` to return a dynamic task name via URL. 8. Create a route `/task/{name}/{priority}` to accept multiple parameters. 9. Create a Blade view `tasklist.blade.php` to display task names. 10. Use Blade `{{ }}` syntax to display values dynamically in the view.
2	**Blade Template Engine & View Management (Tasks 11–20)**
	11. Pass data from route to view using `with()` method. 12. Use `compact()` to pass multiple task values to the view. 13. Create a master layout `layouts.master.blade.php` using `@yield`, `@section`, and `@extends`. 14. Create reusable header and footer using `@include`. 15. Push custom texts or messages using `@push` and render with `@stack`. 16. Display tasks dynamically using `@foreach`. 17. Use `@if` to check if task is completed. 18. Use `@unless` to show a message when no tasks are present. 19. Use `@switch` to change display based on task status. 20. Create 5 sample tasks using `@for` loop in the view.
3	**Forms, CSRF & Validation (Tasks 21–30)**
	21. Design a form to add a new task (title and description). 22. Use `@csrf` to protect the form from cross-site request forgery. 23. Create a POST route `/tasks/store` to accept the form input. 24. Create a controller `TaskController` and define `store()` method. 25. Validate inputs using `$request->validate(['title'=>'required'])`. 26. Display validation errors using `@error('title')` and `$message`. 27. Show all validation errors using `$errors->all()` in the form view.

	28. Use `$errors->any()` to display a general validation alert. 29. Use `session()->flash('success')` to show a success message. 30. Redirect back to the form using `return redirect()->back()` after submission.
4	**Controllers, Route Management & Session (Tasks 31–40)**
	31. Create a `TaskController` using Artisan (`php artisan make:controller TaskController`). 32. Move route logic to controller's `index()` and `store()` methods. 33. Display all tasks from an array in `index()` method. 34. Create a route `/tasks/create` to display form using `create()` method. 35. Use `store()` method to save a new task in session. 36. Create a named route for storing tasks and link to it from form. 37. Use route-model binding (demo with array) to display a specific task. 38. Create a `show($id)` method to return a task by index. 39. Protect task creation with dummy middleware (auth simulation). 40. Define resourceful routes using `Route::resource('tasks', TaskController::class)`.
5	**Database, Models & Relationships (Tasks 41–50)**
	41. Create a migration file for `tasks` table with `title`, `description`, `status`. 42. Set database connection in `.env` file. 43. Run migration using `php artisan migrate`. 44. Create a `Task` model using `php artisan make:model Task`. 45. Update `store()` method to save tasks in database using Eloquent. 46. Update `index()` method to fetch all tasks from DB using `Task::all()`. 47. Create an edit form to update tasks using `edit()` and `update()` methods. 48. Add delete functionality using `destroy()` method and form with `DELETE` method. 49. Add `user_id` column in tasks table and set up foreign key. 50. Establish Eloquent relationships: `User hasMany(Task)` and `Task belongsTo(User)`.

Expected Outcome of this project/practical:

By the end of this case study, students will have:
- Installed and configured Laravel
- Built a working To-Do List app
- Learned Laravel's routing, views, controllers, Blade, validation, sessions, migrations, models, and Eloquent relationships
- Gained practical understanding of building CRUD apps with Laravel

Text Book:

1) Matt Stauffer, "LARAVEL Up and Running, A framework for building modern PHP Apps", 3rd Edition, O'REILLY

Reference Books:

1. Martin Bean, "Laravel 5 Essentials", Packet Publishing
2. Fernando Monteiro, "Hands-On Full-Stack Web Development with Angular 6 and Laravel 5 ", Packt Publishing
3. Web Technologies : HTML,CSS3, JAVASCRIPT, jQUERY, AJAX, PHP,XML, MVC and LARAVEL), Black Book, 2018, Dreamtech

Web Resources:

- Online Laravel 5.2 Documentation (https://laravel.com/docs/5.7)
- Nathan Wu, Learning Laravel 5 Cookbook (https://learninglaravel.net/laravelbook)
- https://laravel-news.com/category/laravel-tutorials
- https://laravel.com