Course Description

M. Tech. In Computer Science and Engineering

Title of Course: Advanced Computer Networks **Course Code: CS501**
L-T Scheme: 3-0 **Course Credits: 3**

Prerequisite: Students must have knowledge of Programming, Operating System, and Computer Networks.

Objectives: To be able to design, configure, and edit network structures and protocols.

Learning Outcomes: After completion of this student will be able to:

Course Outcome	Description
CO1	Comprehend the responsibility of each layer of TCP/IP reference model.
CO2	Describe issues in internet protocols.
CO3	Demonstrate understanding of advanced routing protocols.
CO4	Apply congestion control methods and to provide solution to related issues.
CO5	Analyze MPLS, VPN, and SDN protocols.

Course Contents:

Unit-I: Reference models the Open system interconnect and the TCP/IP. Need of protocols at each layer.
Unit-II: IPv4 and IPv6 – features, addressing, and transition mechanisms**,**

Unit-III: Routing inside and outside of networks, advanced routing protocols, IGP and EGP. QoS-aware routing: metrics and algorithms.

Unit-IV: TCP Congestion control (AIMD, TCP Reno, TCP Vegas, BBR, etc.), TCP variants and performance tuning, Flow and error control mechanisms, SCTP, QUIC – modern transport protocols, Multipath TCP (MPTCP).

Unit-V: MPLS architecture, label switching, traffic engineering. VPNs (Site-to-site, Remote Access), IPSec, GRE tunneling. NAT, PAT, and IPv6 tunneling techniques, Software-Defined Networking (SDN): architecture, OpenFlow. Network Function Virtualization (NFV)

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered upto Test-1
Test-2	25 Marks	Syllabus covered upto Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

References

1. **Computer Networks** by Andrew S. Tanenbaum & David Wetherall.
2. Michael A. Gallo, William M. Hancock, Computer Communication and Networking Technologies, Thomson Press, 2002.
3. Dimitri Bertsekas, Robert Gallager, Data Networks, second edition, PHI, 1992.
4. **Computer Networking: Principles, Protocols and Practice** – Olivier Bonaventure.
5. RFCs – especially for TCP, BGP, MPLS, QUIC, etc.

Course Description

Title of Course: Advanced Database Systems **Course Code: CS502**
L-T Scheme: 3-0 **Course Credits: 3**

Prerequisite: Student must have knowledge about Database Management Systems.

Objectives: To develop the ability to design, implement and manipulate databases as well as to build Database management systems.

Learning Outcomes:

- Ability to build normalized data bases made a secure application for real time systems.
- Ability to design systems by using Models.
- Ability to develop skills of writing applications by using SQL and PL SQL.
- Ability to understand query optimization techniques.
- Understanding of transaction processing.
- Ability to handle recovery and concurrency issues in the database.
- Ability to design web services will be use in different applications.

Course Contents:

Simple database applications and Users: Single user, Multi User, Review of relational databases- EER model, object oriented model, relational algebra, Transforming data models into database design, determination of data requirements for databases. SQL for database processing, advanced SQL, Ranked queries, joins among relations, advanced database manipulation using SQL and PL SQL.

Query Processing and Optimization: Evaluation of Relational Operations, Transformation of Relational Expressions, Indexing and Query Optimization, Data access from disk, Index based access, Sort and Join Processing, Physical plan selection, Limitations of Relational Data Model;

Distributed Query Processing and Optimization: Distributed Transaction Modeling and Concurrency Control, Distributed Deadlock, Commit Protocols, Design of Parallel Databases, Parallel Query Evaluation. Parallel and Distributed Databases: Distributed Data Storage, Fragmentation & Replication, Location and Fragment Transparency.

Advanced Transaction Processing: Nested and Multilevel Transactions, Compensating Transactions and Saga, Long Duration Transactions, Weak Levels of Consistency, Transaction Work Flows, And Transaction Processing Monitors.

Secured Database applications: Database Security - Database security risks, database users and privileges. Authentication, authorization, access control methods, data encryption, Database Recovery- Failure classifications, recovery and atomicity, log based recovery, shadow paging, recovery with concurrent transactions, Buffer management, Multimedia and Mobile databases: Multimedia databases, web-enabled database, temporal databases, Mobile databases.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered upto Test-1
Test-2	25 Marks	Syllabus covered upto Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text Book

1. R. Ramakrishnan, J. Gehrke, Database Management Systems, McGraw Hill, 2004
2. Database System Concepts" – *Abraham Silberschatz, Henry F. Korth, and S. Sudarshan Publisher: McGraw-Hill*

References

1. "Fundamentals of Database Systems" – *Ramez Elmasri and Shamkant B. Navathe Publisher: Pearson.*
2. "Oracle PL/SQL Programming" – *Steven Feuerstein Publisher: O'Reilly Media.*
3. "Learning PHP, MySQL & JavaScript" – *Robin Nixon Publisher: O'Reilly Media.*
4. "Information Security: Principles and Practice" **–** *Mark Stamp Publisher: Wiley.*
5. "Mobile Database Systems" – *Vijay Kumar Publisher: Wiley-Interscience.*
6. "Multimedia Database Management Systems" **–** *B. Prabhakaran Publisher: Springer.*

Course Description

Title: Research Methodology and IPR Code: HS520
L-T-P scheme: 2-0-0 **Credits: 2**

Prerequisite: There is no prerequisite for this course.

Objective: This course is designed to enable students:

1. To identify the issues and concepts salient to the research.
2. To understand the overall processes of designing a research study from its inception to its report.
3. To use basic techniques of quantitative and qualitative research.
4. To gain knowledge to develop data analytics skills and meaningful interpretation to the data sets so as to solve the research problem.
5. To conduct the research work and draft research articles, synopsis and report.

Learning Outcomes:

Course Outcome	Description
CO1	Develop understanding of basic concepts of research and its methodologies, various kinds of research, objectives of doing research, research process, and ethical dimensions of conducting research.
CO2	Select and define appropriate research problem & identify various sources of information for literature review and data collection.
CO3	Outline various concepts of research design, sample design, measurement and scaling techniques, IPR and their application.
CO4	Formulation of research hypotheses and applying various techniques of data analysis for hypothesis testing with the aspects of IPR
CO5	Describe the real-world problems using basic techniques of descriptive and inferential data analysis and to reach a conclusion based on the data analysis, and to make and defend a recommended course of action.

Course Outline

1. Introduction to Research Methodology: Meaning of Research, Objectives of Research, Motivations in Research, Types and Approaches of Research, Significance of Research, Research Methods v/s Methodology, Research and Scientific Methods, Research Process, Criteria of Good Research.

2. Defining the Research Problem: What is Research Problem?, Selecting the Problem, Necessity of and Techniques in defining the problem

3. Research design: Basic research process, Meaning and necessity of research design, Features of good research design, Different research designs, Basic principles

4. Sample Design: Sample survey, Implications of a sample design, Steps in sampling design, Need for sampling, Important sampling distributions, Criteria of selecting a sampling procedure, Characteristic of a good sample design, Different types of samples design

5. Measurement and scaling techniques: Measurement in research, Measurement scales, Sources of errors in measurement, Techniques of developing measurement tools, Scaling, Scale classification bases, important scaling techniques

6. Methods of Data Collection and Presentation: Various methods of Primary and secondary data collection Selection of appropriate method for data collection; Case Study Method, Guidelines for developing questionnaire, successful interviewing. Survey v/s experiment, Methods of Data presentation

7. Processing and analysis of data: Frequency distributions, Measure of central tendency, Measures of dispersion , Simple regression analysis, Multiple correlation and regression, Estimation of population means, Sample size and its determination, Chi- square test, Analysis of Variance and Co-variance

8. Research Ethics: Ethics introduction, ethics in research work, concept of similarity, Plagiarism, Academic Integrity, Departmental Academic Integrity Panel (DAIP), Institutional Academic Integrity Pane (IAIP)

9. **Intellectual property rights:** Intellectual Property – The concept of IPR, Evolution and development of concept of IPR, IPR development process, Trade secrets, utility Models, IPR & Bio diversity, Role of WIPO and WTO in IPR establishments, Right of Property, Common rules of IPR practices, Types and Features of IPR Agreement, Trademark, Functions of UNESCO in IPR maintenance., Trademark, Patents.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1 & Unit-2
Test-2	25 Marks	Based on Unit-3 & Unit-4 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-5 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text Book

- Research Methodology-Methods and Techniques by C.R. Kothari; New Age publication.
- Research Methodology by R.Panneerselvam ; PHI publication.
- Research Methodology by P.K. Sharma
- H.A. Taha. (1992) Operations Research- An Introduction, New York: Macmillan.
- F. S. Hiller and G. J. Liberman. Introduction to Operations Research, San Francisco: Holden- Day.
- Intellectual Property: A Very Short Introduction, Siva Vaidhyanathan
- Intellectual Property: The Law of Trademarks, Copyrights, Patents, and Trade Secrets" Deborah E. Bouchoux

Course Description

Title of Course: High Performance Computer Architecture **Course Code: CS503**
L-T Scheme: 3-0 **Course Credits: 3**

Prerequisite: Student must have knowledge about the hardware and software component about the computer and basic of computer architecture and parallel computing.

Objectives: The main aim of this course is intended to develop, implement, and debug assembly language programs that meet stated specifications. To explain bus transactions, memory organization and address decoding, basic I/O interfaces and port addressing. To understand how to control components of a computer system through the use of hardware and software interrupts.

At the conclusion of the course, following learning objectives are expected to be achieved:

1. You will broaden your knowledge of computer system organization and Architectures.
2. You will gain knowledge in technical aspects of computer system design.
3. You will gain understanding of computer arithmetic both integer and floating point.
4. You will acquire the background for understanding next-generation CPUs.
5. You will learn a computer programming model at a level that enables you to write assembly language programs for the processor meeting given specifications.

Learning Outcomes: The students shall acquire the generic skills to know about the all aspect of computer like working of memory, memory calculations, design and implement of computer architecture along with analysis of practical aspects.

Course Outcome	Description
CO1	Implement and simulate VHDL models using behavioral, dataflow, structural, and mixed modeling approaches.
CO2	Analyze the functional organization of computer systems, including pipelining, RTL, memory hierarchy, and performance metrics.
CO3	Evaluate modern performance enhancement techniques such as branch prediction, dynamic scheduling, and multithreading.
CO4	Apply and assess multiprocessing and parallel architectures including Amdahl's Law, cache coherence, and GPU-based computing.
CO5	Explain and compare distributed computing paradigms including grid, cluster, and cloud computing systems.

Course Contents:

- **Introduction to VHDL:** Implement basic VHDL constructs, Implement modeling structures of VHDL: Behavioral, dataflow, structural, mixed design, use VHDL building blocks: entity architecture, subprograms, package, package declarations, package body, test –Bench, State machine modeling, fault analysis and hazard detection.
- **Functional Organization:** Review the concepts of Computer Architecture – RTL, Micro program, Pipelining and ILP, Memory and I/O. Processor and storage hierarchy, system performance, Performance – Benchmarks, Metrics and their Limitations, Fault tolerance, pipelining timing analysis, and area performance analysis.
- **Performance Enhancements:** Branch Prediction, Dynamic scheduling, Speculative Execution (software/hardware), Superscalar Architecture, Out-of-order Execution, and Multithreading, VLIW and EPIC Architectures, Power Aware Computing, SPEC Mark vs. DTMR cache performance.
- **Multiprocessing:** Amdahl's Law, Multicore and Multithreaded Processors, Flynn's Taxonomy: Multiprocessor Structures and Architectures, Memory Synchronization and Cache Coherence, Interconnection Networks, Programming Multiprocessor Systems, GPU and Special-Purpose Graphics Processors. Case Study – Blue Gene, Road Runner
- **Distributed Systems:** Grids, Cluster Computing, Cloud Computing

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered upto Test-1
Test-2	25 Marks	Syllabus covered upto Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

References

1. Patterson D A, Hennessy J L, "Computer Architecture – A Quantitative Approach" Elsevier, 3rd Edition.
2. Kai Hwang, "Advanced Computer Architecture – Parallelism, Scalability, Programmability", Mc Graw Higher Education, 1992.

3. Michael J. Flynn, "Computer Architecture-pipelined and parallel processor design", Narosa publication, 8th Reprint, 2008.
4. J. E. Smith, "Characterizing Computer Performance with a Single Number", Communications of ACM, Volume 31, Issue 10 (October 1988), Pages: 1202 – 1206, 1988.
5. D. W. Wall, "Limits of Instruction Level Parallelism", Architectural Support for Programming Languages and Operating Systems, Santa Clara, Pages: 176 – 188, 1991.

Course Description

Title of Course: Software System Lab-I **Course Code: CS601**
L-T-P scheme: 0-0-2 **Course Credit: 1**

Prerequisite: Students must have good knowledge of C++/JAVA.

Objectives: To develop an understanding of the various components of the UNIX/Linux operating systems from a system programmer's perspective including both the shell and programming interfaces. To develop the ability to use a NS-2 simulator for a designed network.

Learning Outcomes: Students will be able to:

1. Efficient in system programming
2. Creating a process and making a IPC connection
3. Socket programming and System Call functions
4. Comfortable in Network Simulator

Course Contents:

Introduction to UNIX commands and Shell programming, Process model, Process environment, Process creation and termination, Interprocess communication, Process synchronization, Pipes, Named pipes, Introduction to assembly language, Introduction to Compiler, Introduction to NS-2, Mini Project.

Note: There will be three minis Project and each project will be done by different groups.

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Based on Lab Exercises: upto P1**
P-2		15 Marks	**Based on all Lab Exercises**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

References

1. Kay A. Robbins, Steven Robbins, "UNIX Systems Programming: Communication, Concurrency, and Threads", Prentice Hall 2003.
2. Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, "Compilers: Principles, Techniques, and Tools", Pearson Education 2003.
3. Jeff Duntemann, "Assembly Language Step-by-Step: programming with DOS and Linux" Wiley Dreamtech India Pvt. Ltd., New Delhi, 2003.
4. Arnold Robbins, "UNIX in a Nutshell", O'Reilly 4th edition.

5. David I. Schawartz,"Introduction to UNIX", Prentice Hall, second edition .

Title of Course: Artificial Intelligence in Manufacturing **Course Code: CS701**
L-T-P Scheme: 3-0-0 **Course Credits: 3**

Course Objectives:

- Introduce the fundamentals of AI and its importance in manufacturing.
- Explore machine learning, robotics, and knowledge-based systems as applied to manufacturing.
- Equip students with the ability to analyze manufacturing problems and propose AI-based solutions.
- Familiarize students with Industry 4.0 technologies integrated with AI for smart factories.

Course Outcomes:

Course Outcomes	Description
CO1	Understand the fundamentals of AI and its relevance to manufacturing.
CO2	Apply machine learning techniques in manufacturing contexts.
CO3	Analyze and design AI-based solutions for automation and optimization of manufacturing processes.
CO4	Evaluate case studies and current practices of AI in smart manufacturing.
CO5	Appreciate the role of AI in Industry 4.0 and future manufacturing trends.

Course Contents:

Unit 1: Introduction to AI in Manufacturing

Evolution of manufacturing systems, Introduction to AI and Industry 4.0, AI technologies in smart manufacturing

Unit 2: Knowledge-based Systems in Manufacturing

Expert systems and decision support, Knowledge representation and reasoning, Applications in process planning and quality assurance

Unit 3: Machine Learning for Manufacturing

Supervised and unsupervised learning algorithms, Predictive maintenance and defect detection, Optimization of manufacturing processes using ML

Unit 4: Robotics and Intelligent Automation

AI in industrial robots, Computer vision applications, Human-robot collaboration in manufacturing environments

Unit 5: Advanced Applications and Case Studies

AI in supply chain and inventory management, Digital twins and simulation, Case studies on smart factories, predictive maintenance

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered upto Test-1
Test-2	25 Marks	Syllabus covered upto Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Textbook:

1. Srinivas, K., & Anandakrishnan, V. (2020). Artificial Intelligence for Industrial Applications. Springer.

Reference Books:

1. Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th Edition). Pearson.
2. Lee, J., Bagheri, B., & Kao, H.-A. (2015). "A Cyber-Physical Systems Architecture for Industry 4.0-based Manufacturing Systems", Manufacturing Letters, Elsevier.
3. Li, B., Hou, B., Yu, W., Lu, X., & Yang, C. (2017). "Applications of Artificial Intelligence in Intelligent Manufacturing: A Review", Frontiers of Information Technology & Electronic Engineering.
4. Zhang, X. (2019). Machine Learning Applications in Nontraditional Manufacturing Processes. Springer.

Course Description

Title of Course: Advanced Algorithm Lab **Course Code: CS805**
L-T-P scheme: 0-0-2 **Course Credit: 1**

Prerequisite: Good knowledge of Computer programming, Data Structures and algorithms. Proficiency in a high-level programming language is required.

Objectives:

- To practically implement advanced algorithms and data structures.
- To strengthen the ability to apply algorithm design paradigms such as greedy programming and dynamic programming to solve problems.
- To empirically analyze the performance and correctness of algorithms.
- To gain hands-on experience with algorithms from various domains, including computational geometry and cryptography.

Course Outcomes:

Course Outcomes	Description
CO1	Implement and evaluate the time complexity of various sorting and searching algorithms.
CO2	Develop and code solutions for optimization problems using Greedy and Dynamic Programming techniques.
CO3	Implement standard algorithms for string matching and computational geometry.
CO4	Create programs that use basic cryptographic and primarily testing algorithms.

Course Contents:

The lab will consist of programming assignments based on the topics covered in the "Advanced Algorithms Lab" course. Students will be expected to implement the following:

Sorting & Searching Algorithms:

- Implementation and empirical performance analysis of Quick Sort and Merge Sort.
- Implementation of non-comparison based sorting algorithms like Radix Sort and Bucket Sort.
- Implementation of a Hash Table with a chosen collision resolution strategy.

Greedy Algorithms:

- Implementation of Dijkstra's algorithm for the Shortest Path Problem.

- Implementation of Prim's and Kruskal's algorithms for finding a Minimum Spanning Tree.

Dynamic Programming:

- Implementation of a solution for the 0/1 Knapsack problem.
- Implementation of a solution for the Matrix Chain Multiplication problem.

String Matching:

- Implementation of the Rabin-Karp algorithm.
- Implementation of the Knuth-Morris-Pratt (KMP) algorithm.

Computational Geometry:

- Implementation of an algorithm, such as Graham scan, to find the Convex Hull of a set of points.

Cryptographic & Number Theory Algorithms:

- Implementation of the RSA public-key cryptosystem.
- Implementation of the Miller-Rabin primality test as part of the RSA implementation.

Evaluation Scheme:

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Based on Lab Exercises: upto P1**
P-2		15 Marks	**Based on all Lab Exercises**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

References

1. Algorithms: S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani, McGraw-Hill Science/Engineering/Math;
2. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2nd Edition, 2001.
3. Sedgewick, Robert. *Algorithms in C*. 3rd edition. Addison Wesley, 2002.

Course Title: Advanced Database Systems LAB Course Code: CS803

L-T Scheme: 0-0-2 Course Credits: 1

Prerequisite: Basic knowledge of Database Management Systems (DBMS)

Lab Objectives:

1. To provide practical experience in the design and implementation of advanced database concepts.
2. To enhance skills in writing complex SQL and PL/SQL programs.
3. To implement distributed, parallel, and secure database applications.
4. To explore recovery, indexing, and query optimization techniques through experiments.
5. To develop real-world web and mobile-enabled secure database applications

Lab Learning Outcomes:

Course Outcome	Description
CO1	Design and develop secure, normalized, and efficient database systems.
CO2	Write complex SQL, PL/SQL procedures, functions, and triggers.
CO3	Understand and implement indexing, query optimization, and distributed transactions.
CO4	Apply concurrency control and recovery techniques to real-time database environments.
CO5	Develop secure web and mobile-integrated database systems.

Lab Experiments:

Exp. No.	Title of the Experiment
1	**Design and Normalize a Complex Relational Schema** Model a realistic scenario (e.g., university or hospital system) by creating ER diagrams, mapping to relational schema, and normalizing up to 3NF/BCNF to eliminate redundancy.
2	**Advanced SQL Queries: Nested, Ranked, Analytical Functions** Craft advanced SQL using nested and correlated sub queries, window functions such as RANK, ROW_NUMBER, LEAD/LAG, analytic aggregates (SUM, AVG), ROLLUP/CUBE.

3 **Stored Procedures, Functions, and Exception Handling in PL/SQL** Implement procedural business logic with PL/SQL by writing stored procedures, functions, using cursors, handling exceptions (NO_DATA_FOUND, OTHERS), and structuring robust error control.

4 **Triggers for Data Integrity and Auditing** Create BEFORE/AFTER INSERT/UPDATE/DELETE triggers to enforce constraints; maintain audit logs (user, timestamp, old/new values.

5 **Designing Indexes and Query Optimization Analysis** Design primary, secondary, composite, and function-based indexes. Analyze EXPLAIN PLAN outputs, tuning queries using join strategies, cost metrics, and index selectivity

6 **Implementation of Fragmentation and Replication Strategies** Simulate horizontal and vertical fragmentation across distributed sites. Plan and test snapshot, transactional, and merge replication to ensure availability and consistency across nodes.

7 **Distributed Query Processing and Optimization** Develop SQL queries across distributed datasets, incorporate site-selection, join ordering, data localization, and network cost in optimization.

8 **Concurrency Control: Transaction Isolation Levels and Deadlock Simulation** Demonstrate and compare RCC, RC, RR, and Serializable isolation levels. Create and resolve deadlock scenarios using locking, timestamps, and multi-granularity techniques.

9 **Failure Recovery and Log-based Recovery Simulation** Simulate system crashes, apply UNDO/REDO recovery using Write-Ahead Logging. Understand ARIES-style checkpoint protocols to restore ACID properties

10 **Secure Database Access with Role-based Authorization** Define roles and grant privileges (SELECT, INSERT, EXECUTE) to roles/users. Set up schemas and enforce fine

11 **Development of a Web-based Secure Database Application (Mini Project –1)**
Build a secure front-end (HTML/Java/JDBC or Python/Flask) and back-end DB (Oracle/MySQL/Postgres). Implement form-based CRUD operations, parameterized queries, authentication, and session managemen

12 **Integration with Mobile/Cloud-based Storage (Mini Project -2)**

Evaluation Scheme:

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Based on Lab Exercises: upto P1**
P-2		15 Marks	**Based on all Lab Exercises**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	

Total	**100 Marks**

Lab Software & Tools Required:

- Oracle Database 11g/12c or higher
- PostgreSQL / MySQL Server
- PL/SQL Developer / pgAdmin / MySQL Workbench
- Apache/PHP or Python Flask/Django for web interface
- Java or Kotlin (optional) for mobile interfacing
- Firebase / REST API tools for mobile DB connectivity
- Git for version control

Lab References:

1. Steven Feuerstein, *Oracle PL/SQL Programming*, O'Reilly Media
2. Robin Nixon, *Learning PHP, MySQL & JavaScript*, O'Reilly Media
3. Mark Stamp, *Information Security: Principles and Practice*, Wiley
4. Ramez Elmasri & Shamkant Navathe, *Fundamentals of Database Systems*, Pearson
5. Vijay Kumar, *Mobile Database Systems*, Wiley-Interscience