Pokhara University
Faculty of Science and Technology

Course Code: CMP 323 (3 Credits)	Full Marks: 100
Course Title: Software Engineering (3-0-1)	Pass Marks: 45
Nature of the Course: Theory/Practical	Total Lectures: 48 hours
Level: Bachelor | Year: II / Semester: IV	Program: Bachelor of Computer Application

1. Course Description:

This course introduces students to essential principles and methodologies of software engineering, focusing on Object-Oriented Analysis and Design (OOAD) using Unified Modeling Language (UML). It covers software development process models, software design, coding practices, software testing, maintenance, and quality assurance. Students will apply these concepts through case studies and project work, ensuring practical skills in software design and development.

2. General Objectives:

1) To provide students with a comprehensive understanding of the principles, methodologies, and best practices in software development.
2) To equip students with essential skills in Object-Oriented Analysis and Design (OOAD) using UML, software process models, coding standards, software testing, and software quality assurance.
3) To design, develop, and maintain software systems that meet high standards of functionality, quality, and performance in real-world scenarios.

3. Methods of Instructions:

1. Lecture and discussion
2. Practical
3. Demonstration
4. Presentation
5. Case study

4. Course Contents:

Specific Objectives	**Contents**
Unit 1: Introduction to Software Engineering	**5 hours**
• Describe concepts and challenges in software engineering. • Analyse the attributes of good software and ethical responsibilities in the software industry.	1.1 Overview of Software Engineering 1.2 Definition and characteristics of software. 1.3 Software engineering as a discipline and its challenges. 1.4 Professional Practices in Software Engineering 1.5 Attributes of good software. 1.6 Software development life cycle (SDLC) and ethical practices.
Unit 2: Software Development Process Models	**4 hours**
• Apply various software development process models to real-world software projects. • Analyse the role of CASE tools in the software development life-cycle, software process, software process model:	2.1 Software Process Models 2.2 The Waterfall Model, Evolutionary Development, Component Based Software Engineering(CBSE), Process Iteration, Incremental Delivery, Spiral Development, Rapid Software Development, Agile Methods, Extreme Programming, Rapid Application Development and Rational Unified Process (RUP). 2.3 Overview and classification of CASE tools. 2.4 Role of CASE tools in modern software engineering practices.
Unit 3: System Analysis in OOAD	**4 hours**
• Analyze and document system requirements in an object-oriented context.	1. Requirements in Object-Oriented Systems 2. Identifying functional and non-functional requirements in object-oriented systems.

• Develop use case models to define functional requirements in OOAD.	3. Use of Use Case Diagrams to model system requirements. 4. Requirement Modeling in OOAD 5. Creating Use Case Descriptions and Use Case Diagrams in UML for capturing functional requirements.
Unit 4: Object-Oriented Design (OOD) with UML	**12 hours**
• Develop models of software systems using UML diagrams. • Apply object-oriented principles and design patterns in software system design	4.1 Introduction to OOD 4.2 Design Concept; Abstraction, Architecture, Patterns, Modularity Cohesion, Coupling Information Hiding, Functional Independence, Refinement 4.3 Object-oriented principles: abstraction, encapsulation, inheritance, polymorphism. 4.4 Benefits of OOAD in software engineering. 4.5 Class diagrams, object diagrams, and sequence diagrams, State machine diagrams, component diagrams, and deployment diagrams. 4.6 Introduction to common design patterns such as Singleton, Factory, Observer. **4.7** Procedural Design Using Structural Methods, User Interface Design, Human Computer Interaction, Information Presentation, Interface Evaluation, Design Notation
Unit 5: Software Testing and Quality Assurance	**9 hours**
• Compare different software testing techniques to ensure software quality. • Implement verification, validation, and quality assurance practices in software projects.	5.1 Testing Techniques 5.2 Unit testing, integration testing, and system testing. 5.3 Black-box and white-box testing techniques. 5.4 Verification and Validation 5.4 The importance of software testing in the development life cycle. 5.5 Software Quality Assurance (SQA) 5.6 Overview of software quality standards (ISO, CMM). 5.7 Importance of software quality assurance (SQA) processes. 5.8 Risk Categories, The RMM Plan
Unit 6: Coding Practices and Programming	**4 hours**
• Analyse the best coding practices to improve software quality and maintainability. • Compare appropriate programming languages and tools for developing software systems.	6.1 Programming Languages and Development Tools 6.2 Selecting programming languages and tools for software development. 6.3 Best Practices in Coding 6.4 Clean code practices, code readability, and refactoring. 6.5 Introduction to software development environments and integrated development environments (IDEs).
Unit 7: Software Maintenance and Evolution	**6 hours**
• Describe types of software maintenance and their significance in the software life-cycle. • Apply re-engineering and configuration management techniques to maintain and improve software systems.	7.1 Software Maintenance 7.2 Evolving Nature of Software, 7.3 Types of software maintenance: corrective, adaptive, perfective, preventive. 7.4 Re-engineering and Configuration Management 7.5 Process of software re-engineering and version control. 7.6 Importance of configuration management in large-scale software systems.

5. List of Practical:

Lab

1. Requirements Gathering and Use Case Modeling: Conduct a requirements gathering session to identify functional requirements for a system. Create Use Case Diagrams and develop use case descriptions for the identified requirements.
2. Class Diagram Design: Design a Class Diagram for a given system based on the collected requirements. Model classes, attributes, methods, and relationships (inheritance, association, aggregation).
3. Sequence Diagram Design: Create a Sequence Diagram to represent the interaction between objects in a use case. Model the flow of messages and interactions between system components.
4: Activity Diagram for Workflow Modeling: Design an Activity Diagram to represent the flow of activities in a business process. Capture the flow of control, decision points, and concurrency in the system.
5: Coding Best Practices and Refactoring: Implement a software module using clean code practices. Perform code refactoring to improve readability, maintainability, and performance of an existing module.
6: Unit Testing with JUnit: Write Unit Tests for a given software module using JUnit. Perform testing on individual components and analyze the test results.
7: Software Version Control with Git: Set up a project in Git and implement version control for the software development process. Use Git commands to track changes, commit, push, pull, and merge code.
8: Software Architecture Design: Design a multi-layered Software Architecture (e.g., client-server or layered architecture) for a system. Create Component Diagrams to visualize the system's architecture.
9: Software Quality Assurance (SQA) Techniques: Conduct a quality assurance review for a software project. Use static code analysis tools and apply software quality standards such as ISO or CMMI.
10: Black-Box and White-Box Testing: Perform Black-Box Testing and White-Box Testing on a software application. Design test cases, execute tests, and document results to ensure software quality.

6. Evaluation System and Students' Responsibilities:

6.1 Evaluation System:

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class participation, quizzes, assignments, presentations, written exams, etc. The tabular presentation of the evaluation system is as follows.

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		**30**	**Semester End examination**	**50**
Attendance / Class Participation	10%			
Assignments	20%			
Project Work/Presentations	10%			
Term Exam	60%			
Practical		**20**		
Attendance and Lab Participation	10%			
Lab Report	20%			
Lab Examination	40%			
Viva Examination	30%			
Total Internal Marks		**50**		
Full marks=50+50 =100				

6.2 Students' Responsibilities:

Each student must secure at least 45% marks separately in internal assessment and practical evaluation with 80% attendance in the class in order to appear in the Semester End Examination. Failing to get such a score will be given NOT QUALIFIED (NQ) to appear for the Semester-End Examinations. Students are advised to attend all the classes, formal exam, test, etc. and complete all the assignments within the specified time period. Students are required to complete all the requirements defined for the completion of the course.

7. Prescribed Books and References:

Text Books

1. Roger S. Pressman, "Software Engineering: A Practitioner 's approach", 6th Edition, McGraw Hill International edition,2005

Reference Books

1. Ali Behforooz , and Frederick J. Hudson, "Fundamental of Software Engineering", OUP,1996
2. Ian Somerville, "Software Engineering", 9th Edition Addison-Wesley, 2010.
3. Rajib Mall," Fundamental of Software Engineering", 3rd Edition, 2010.

Pokhara University Faculty of Science and Technology	
Course Code.: CMP 226 (3 Credits)	Full Marks: 100
Course Title: Database Management System (3-0-3)	Pass Marks: 45
Nature of the Course: Theory/Practical	Total Lectures: 48 hours
Level: Bachelor | Year: II / Semester: IV	Program: Bachelor of Computer Application

1. Course Description:

This course is designed to encompass the fundamental concepts of database management system. These concepts include the concept of database design, database languages and the database system implementation. This course presents the introductory concepts of database security, query processing and optimization, transactions and their concurrency control and recovery policies. This course also introduces the emerging new trended databases such as NoSQL and Block-chain databases. After completion of this course, students can design and implement a database system to develop a software application.

2. General Objectives:

The general objectives of this course are as follows:

- To acquaint the students with the knowledge of relational database design using ER Model.
- To develop the skills in students to design normalized relational database required for a specified application.
- To acquaint the students with the knowledge of database security, query processing and optimization, files and record organizations, transaction, concurrency control, data recovery mechanisms.
- To acquaint the students with concepts of NoSQL databases and Block-chain Databases.

3. Methods of Instructions:

- Lecture and discussion
- Practical
- Project and presentation

4. Course Contents:

Specific Objectives	Contents
Unit 1: Introduction	**7 hours**
• Familiarize with basic concepts of database systems, and its architecture. • Explain the concept for database components and their roles	1. Database Management System purpose and applications, 2. Database Systems vs File Systems, Storage Management and Indexing 3. View of Data- Data Abstraction (Physical, logical and view level, Data Independence) 4. Instances and Schemas, 5. Database Languages (DDL, DML and DCL) 6. Database and Application Architecture- Database System Architecture and Database Application Architecture (two-tier and three-tier)
Unit 2: ER and Relational Model	**8 hours**
• Familiarize with ER Model and Relational Model and relational algebra. • Design the relational database using ER model.	1. Introduction to ER Model: 2. Entity sets, attributes and values, Relationship sets-participation, entity's role, descriptive attributes, degree of relationship set, Mapping Cardinalities, Attributes-simple, composite, single-valued, multi-valued, derived, Entity-Relationship (ER) Diagram, Specialization, Generalization, and Aggregation 3. Key and its types 4. Reducing ER diagrams to Relational Schema **5.** Structure of Relational Databases, Database Schema, Schema Diagrams **6.** Relational Algebra

Unit 3: Structured Query Language	6 hours
• Implement and write DDL and DML queries in the SQL.	1. Structured Query Language (SQL)- SQL DDL and DML 2. Basic Structure of SQL Queries, DDL queries, Basic Operations (Rename, String, Attribute Specification in the select clause, order by, where-clause), Set Operations, Null values, Aggregate Functions, 3. Nested Queries, 4. Join Expressions (Natural Join, Join Conditions, Outer Joins), Views, stored Procedures
Unit 4: Relational Database Design	**6 hours**
• Apply the integrity constraints to implement database securities. • Analyze the database to a defined normal form.	1. Integrity constraints- Domain Constraints, Entity Integrity Constrains, Referential Integrity Constraints, Assertions and Triggers 2. Features of Good Relational Designs 3. Functional dependencies and Armstrong's Axioms 4. Closure of a Set of Functional Dependencies and Closure of Attribute Sets 5. Database Normalization and Normal Forms- 1NF, 2NF, 3NF and BCNF 6. De-normalization for Performance
Unit 5: Query Processing and Optimization	**5 hours**
• Explore the mechanism of query processing and need of query optimization. • Familiarize with different file organization methods.	1. Introduction to Query Processing 2. Equivalence of Expressions 3. Query Cost Estimation 4. Query evaluation and execution plan 5. Query Optimization
Unit 6: Transactions and Concurrency Control	**5 hours**
• Implement ACID properties, isolation, serial schedule. • Familiarize with the need of Concurrency control.	1. Transaction Concepts 2. Transaction Model and State Diagram 3. ACID properties of transaction 4. Serializability- conflict and view serializability 5. SQL Standard Isolation Levels 6. Concurrency Control- Lock-Based Protocols
Unit 7: Security and Crash Recovery	**6 hours**
• Describe the recovery algorithms and techniques to protect and recover the data from various failures. • Explain the need of database security. • Compare the role of access control, authorization, views and encryption mechanism to provide database security.	1. Security and integrity violations 2. Access control and authorization, views 3. Failure classification 4. Recovery and Atomicity- log records, database modification, concurrency control and recovery, transaction commit, Redo and Undo Transactions using Log, Check Points, 5. Recovery Algorithm Using Log Records- Transaction Rollback, Recovery after a System Crash, Optimizing Commit Processing
Unit 8: Emerging Trend in Databases	**5 hours**
• Explain the concepts of NoSQL, ODBMS, Distributed Databases	1. NoSQL Databases- Characteristics, Categories, Advantages 2. Object Oriented Database and ORM 3. Parallel and Distributed Databases 4. Data Warehouse and Data Mining

5. Practical Works

Laboratory work should cover the database design, use of database languages and database system implementation using any relational database management system such as MS SQL or MySQL or Oracle etc. Students should complete the following tasks in laboratory:

SN	Tasks to Complete
1	Introduction to MS SQL (or any RDBMS), its data types and its installation.
2	All SQL DDL operations studied in Unit 2 such as creating database, creating tables, delete database, drop table, alter etc.
3	All SQL DML operations studied in Unit 2 such as database modification operation- insert into, delete, update etc.
4	Implementing Join Expressions (Natural Join, Join Conditions, Outer Joins)
5	Implementing Stored Procedures.
6	Illustration and implementation of Views.
7	Implementing Integrity constraints (Domain Constraints, Entity Integrity Constrains, Referential Integrity Constraints)
8	Implementing Assertions and Triggers.
9	Implementation of Database Security and Privileges: Grant and Revoke Commands, Commit and Rollback Commands.
10	Connecting database with connection string using any standard programming language and executing SQL queries.

Students should submit a project work that uses all the knowledge obtained from this course to design and implement a database system for any application that students chose. The students should design the database using ER model and present using ER diagram which are then reduced to relational schema. The students should apply the constraints studied in this course including triggers. The database should be in at least 3NF. The marks for the practical evaluation must be based on the project work submitted by students.

6. Evaluation System and Students' Responsibilities:

6.1 Evaluation System:

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class participation, quizzes, assignments, presentations, written exams, etc. The tabular presentation of the evaluation system is as follows.

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		**30**	**Semester End examination**	**50**
Attendance / Class Participation	10%			
Assignments	20%			
Project Work/Presentations	10%			
Term Exam	50%			
Practical		**20**		
Attendance and Lab Participation	10%			
Lab Report	20%			
Lab Examination	40%			
Viva Examination	30%			
Total Internal Marks		**50**		
Full marks=50+50				

6.2 Students' Responsibilities:

To be eligible for the Semester End Examinations, each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the Semester End Examination. Failing to obtain such score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term

examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, quiz, test, etc. there won't be any provision for a re-exam.

7. Prescribed Books and References:

Books

1. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). *Database system concepts.* McGraw Hill

References

1. Majumdar, A. K., & Bhattacharyya, P. (1996). *Database Management Systems*. McGraw Hill.

2. Elmasri, R., & Navathe, S. B. (1994). Fundamentals of Database Systems Benjamin Cummings. *Redwood City, CA.*

3. Everest, G. C. (1986), *Database management*. McGraw-Hill, Inc.

Pokhara University Faculty of Science and Technology	
Course Code.: CMP 242 (3 Credits)	Full Marks: 100
Course Title: Computer Graphics and Multimedia Technology (3-0-2)	Pass Marks: 45
Nature of the Course: Theory/Practical	Total Lectures: 48 hours
Level: Bachelor | Year: II / Semester: VI	Program: Bachelor of Computer Application

1. Course Description:

This course is designed to encompass the fundamental concepts of Computer Graphics. It covers Graphics hardware, various drawing algorithms, animation and multimedia at the industry level. It includes areas such as 2D and 3D modeling, rendering, animation, image processing, virtual reality, and augmented reality. Multimedia, on the other hand, refers to the integration of different forms of media, such as text, graphics, audio, video, and animation, to create interactive and immersive experiences

2. General Objectives:

The general objectives of this course are as follows:

- To introduce the hardware and concept of object rendering.
- To introduce the concept of 2D and 3D Graphics Algorithm
- To familiarize students with the knowledge of multimedia and animation.

3. Methods of Instructions:

- Lecture
- Tutorial
- Practical and case studies

4. Course Contents:

Specific Objectives	Contents
Unit 1: Introduction to Graphics	**6 hours**
• Familiarize with the basic Graphics hardware and interfacing and software	1.1 Application of computer graphics 1.2 CRT monitor working 1.3 Display technology (Raster and Random) 1.4 Color generation technique in CRT 1.5 Frame buffer organization and video controller 1.6 Recent trend virtual reality and augmented reality
Unit 2: Scan Conversion	**6 hours**
• Implement Scan conversion of Graphics Primitive line and Circle.	2.1 Importance of scan conversion 2.2 Line Drawing Method: DDA and BLA 2.3 Mid-point circle generation algorithm
Unit 3: Two Dimensional Transformation and Windows Viewing	**8 hours**
• Implement the function of Basic and Advance transformation of solid body	3.1 2D Geometric Transformation (Translation, Rotation, Scaling) 3.2 Homogeneous Coordinate Representation 3.3 Successive Transformation 3.4 Window to Viewport Transformation 3.5 Cohen Sutherland Line Clipping Algorithm
Unit 4: Three Dimensional Graphics and Hidden Surface Removal	**6 hours**
• Describe the concept of 3D Graphics and 3D object representation	4.1 3D Geometric Transformation 4.2 3D Object Representation- Polygon Surface and

projection and hidden surface removal	Polygon Mesh 4.3 Parallel and Perspective Projection 4.4 Hidden Surface Removal 4.5 Back Face Detection, Z-Buffer Method, Scan Line Method
Unit 5: Illumination Model and Shading	**6 hours**
• Explain the Effect of Light in a computer Scene and Shading techniques	5.1 Basic illumination model 5.2 Specular Reflection and Phong Model 5.3 Polygon Rendering Methods (Flat, Goraud and Phong shading)
Unit 6: Multimedia System and Representation	**9 hours**
• Introduction to Multimedia and media representation and various different kind of authoring tools and Animation	6.1 Introduction to Multimedia 6.2 Different kind of Image Format 6.3 Sound and Audio 6.3.1 Digital Audio Formats 6.3.2 MIDI hardware, software and Messages 6.4 Multimedia Authoring Tools
Unit 7: Multimedia Compression Techniques	**7 hours**
Explain video, compression technology and animation	7.1 Video 7.1.1 Digital video formats (AVI and MOV) 7.2 Compression techniques for audio and video (spatial and temporal) 7.3 Animation and its types

5. Practical Works:

Laboratory work should cover the basic principles of Computer Graphics and Multimedia Systems. It also insists the students to design graphical system with the reference of open source guideline. Students should complete the following tasks in laboratory:

1. To draw a straight line using Digital Differential Analyzer and Bresenham's Line Drawing Algorithm
2. Digitize a circle using mid-point circle Drawing Algorithm
3. Translation, Scaling and Rotation in 2D
4. 2D Screen Viewing Transformation
5. To draw a graphical output primitives using Open GL
6. To implement and test the Image Compression Algorithm (JPEG Compression)

.

6. Evaluation System and Students' Responsibilities:

6.1 Evaluation System:

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class participation, quizzes, assignments, presentations, written exams, etc. The tabular presentation of the evaluation system is as follows.

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		**30**		
Attendance /Class Participation	10%			
Assignments	20%			
Project Work/Presentations	20%			
Term Exam	50%		**Semester End examination**	**50**
Practical		**20**		

Attendance and Lab Participation	20%			
Lab Report	30%			
Lab Examination	30%			
Viva Examination	20%			
Total Internal Marks		**50**		
Full marks=50+50				

6.2 Students' Responsibilities:

To be eligible for the Semester End Examinations, each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the Semester End Examination. Failing to obtain such score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, quiz, test, etc. there won't be any provision for a re-exam.

7. Prescribed Books and References:

Text Books

1. Hearn D & Baker M.P (1997) Computer Graphics, C Version. Pearson Education India
2. Buford, J.F.K(2002), Multimedia Systems, Pearson, India

References

1. Malay K. Pakhira Computer Graphics Multimedia and Animation ,Second Edition PHI Publication

Pokhara University
Faculty of Science and Technology

Course Code.: MTH 320 (3 Credits)	Full Marks: 100
Course Title: Probability and Statistics (3-0-0)	Pass Marks: 45
Nature of the Course: Theory	Total Lectures: 48 hours
Level: Bachelor | Year: II / Semester: IV	Program: Bachelor of Computer Application

1. Course Description:

This course aims to provide Bachelor of Computer Application (BCA) students with a solid understanding of applied statistics, as well as practical skills. It covers statistical concepts including data presentation, measures of central tendency, dispersion, correlation and regression analysis, probability, probability distributions, sampling, estimation, and hypothesis testing. The course focuses on hands-on learning, with laboratory sessions where students can use statistical analysis software tools. Having a mastery of these statistical tools is essential for creating effective dashboards in computer applications, software, web development, and research projects.

2. General Objectives:

The general objectives of this course are as follows:

- To provide students with a fundamental understanding of Statistics, Probability, and Sample Survey techniques, particularly in relation to Computer Applications.
- To equip students with the ability to compute various statistical measures of central tendency and dispersion for given data sets.
- To develop students' skills in utilizing statistical tools and techniques that are relevant to computer applications and research work.
- To familiarize students with statistical software used for data analysis, such as Microsoft Excel, SPSS, or R Studio.

3. Methods of Instructions:

- Lecture and discussions
- Demonstration
- Practical
- Case study

4. Course Contents:

Specific Objectives	**Contents**
Unit 1: Introduction	**3 hours**
• Acquaint fundamental concepts of statistics, its broad applications in various fields. • Familiarize different types of variables and data. • Equip with the understanding of key elements of statistical analysis.	1.1 Introduction to Statistics, 1.2 History and its scope of statistics 1.3 Variable and data, types of variables (quantitative and categorical), types of data (quantitative and qualitative), measurement and scaling, types of scale (nominal, ordinal, interval and ratio), fundamental elements of statistical analysis.
Unit 2: Data Collection	**5 hours**
• Enable in distinguishing between primary and secondary data sources and understand their uses in research. • Acquaint with the fundamental principles and approaches of experimental and survey research methods.	2.1 Sources of Data (Primary and Secondary) 2.2 Primary data collection methods (Field survey, Observation, Experiment) and tools (Questionnaire, Observation Checklist. 2.3 Introduction to Experimental research and survey research 2.4 Data preparation - editing, coding, and transcribing

Unit 3: Presentation of Data	**4 hours**
• Enhance proficiency in organizing and presenting data through the utilization of diverse tabular formats using computer software. • Equip with the skills to effectively visualize data using different graphical methods both manually and using computer software.	3.1 **Tabular Presentation:** Steam-and-leaf display, frequency distribution, relative frequency distribution, cumulative frequency distribution, bivariate frequency distribution. 3.2 **Data Visualization**: Bar diagram (simple, multiple, sub-divided, percentage), pie chart, histogram, frequency polygon, frequency curve, ogive and time-plots.
Unit 4: Summary Measures	**5 hours**
• Enable to compute and interpret various measures of central tendency using computer software. • Develop ability to compute and interpret measures of variation both manually and computer software. • Familiarize and calculate with the concepts of distribution shape	4.1 **Measure of central tendency**: mean (Arithmetic, Geometric and Harmonic), median, mode and mid-hinge 4.2 **Partition Values:** Quartiles, Deciles, Percentiles 4.3 **Measures of variation:** range, interquartile range, semi-inter quartile range standard deviation, Variance and coefficient of variations. 4.4 **Shape of the distribution:** five-number summary, box-and-whisker plot, concept of skewness and kurtosis.
Unit 5: Probability	**5 hours**
• Familiarize with fundamental probability rules and apply them in different scenarios of problem solving. • Make knowledgeable about the concept of conditional probability and Bayes' Theorem, and to enable to apply these concepts in problem-solving.	5.1 **Introduction of Probability:** Basic concepts, Counting rules, Objective and subjective probability, Marginal and joint probability 5.2 **Basic Probability:** Addition rule, Multiplication rules, 5.3 **Conditional Probability:** Concept of Conditional Probability, Bays' Theorem
Unit 6: Discrete Probability Distribution	**10 hours**
• Distinguish between discrete and continuous random variables. • Familiarize with probability distributions. • Explain Binomial, Poisson, and Normal distributions, including the normal approximation of binomial and Poisson distributions.	6.1 **Random variables**: Introduction to random Variable, types of random variable (discrete and continuous), Mean and standard deviation of discrete random variables, Mathematical expectation of discrete random variable 6.2 **Probability distribution:** Concept of probability distribution, types of probability distribution (discrete and continuous), Probability mass function, Probability density function, probability distribution function, Binomial distribution, Poisson distribution and Normal distribution, Normal approximation of binomial and Poisson distribution
Unit 7: Sampling and Estimation of Population Parameters	**5 hours**
• Acquaint with the concepts of population and sample, and sampling methods in statistical analysis. • Calculate and interpret confidence intervals for means and proportions, • Gain hands-on experience using computer software to analyze and interpret data.	7.1 **Sampling:** Population and sample, different types of sampling methods, 7.2 **Estimation:** Law of large numbers, Central limit theorem, point and interval estimation, Statistical confidence, Confidence intervals, Confidence interval estimation for mean and proportion.
Unit 8: Hypothesis Testing	**6 hours**
• Provide with an understanding of hypothesis testing, including steps involved, and the	8.1 **Introduction to Testing of hypothesis: Statistical hypothesis, steps in hypothesis testing,** P-value approach

connection between confidence intervals and hypothesis testing. • Equip with skills single mean and double means using z-tests and t-tests both manually and using computer software. • Acquaint with the Chi-square test for independence and gain practical experience with computer software.	to hypothesis testing, Connection between confidence intervals and hypothesis testing, 8.2 **Test of significance of mean:** Test of single mean, comparing two means (z and t-test) 8.3 **Test of proportion:** Test of single proportion, Comparing two proportions 8.4 **Test of independence using Chi-square test**
Unit 9: Correlation and Regression Analysis	**5 hours**
• Analyze correlations using scatter diagrams, Pearson's correlation coefficient, and Spearman's rank correlation. • Performed simple linear regression analysis using the least squares method to fit models. with Standard Error, and analyze regression output with the help of statistical software.	9.1 **Correlation Analysis**: Scatter diagram, Correlation Coefficient and its Properties, Spearman's rank correlation, Pearson's correlation coefficient. 9.2 **Simple Linear Regression Analysis:** Model fit using least square method, Coefficient of Determination and Standard Error, interpretation of output of regression analysis using statistical software

6. Evaluation System and Students' Responsibilities:

6.1 Evaluation System:

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class participation, quizzes, assignments, presentations, written exams, etc. The tabular presentation of the evaluation system is as follows.

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		50	Semester End examination	50
Attendance / Class Participation	10%			
Assignments	20%			
Project Work/Presentations	10%			
Term Exam	60%			
Full marks=50+50				

6.2 Students' Responsibilities:

To be eligible for the Semester End Examinations, each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the Semester End Examination. Failing to obtain such score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, quiz, test, etc. there won't be any provision for a re-exam.

7. Prescribed Books and References:

Text Book

1. **Levin, Richard I. And David S. Rubin: Statistics for Management, Prentice-Hall of India**

Reference Books

1. Fundamentals of Mathematical Statistics – 1 st Edition S. C. Gupta, V.K. Kapoor and S Chand.
2. Introduction to Probability & Statistics – 4 th Edition J. Susan Milton, Jesse C. Arnold Tata McGraw Hill.
3. Fundamentals of Statistics (7th edition), S C Gupta, Himalaya Publishing house
4. Probability and Statistical Inference (9th Edition), Robert V. Hogg & Elliot Tanis & Dale Zimmerman, Pearson.

Pokhara University
Faculty of Science and Technology

Course Code.: CMP 380 (3 Credits)	Full Marks: 100
Course Title: Web Technology II (3-0-3)	Pass Marks: 45
Nature of the Course: Theory/Practical	Total Lectures: 48 hours
Level: Bachelor | Year: II / Semester: IV	Program: Bachelor of Computer Application

1. Course Description:

This course is designed to give students hands on knowledge of server side development. After completing this course students will not only have theoretical knowledge but also practical, hands-on experience in building, securing, and deploying web applications, making them well-prepared for the challenges of professional web development. The course covers essential topics such as server architecture, web frameworks, database integration, web services, security best practices, and deployment strategies. Students will gain hands-on experience in building, securing, and deploying web applications.

2. General Objectives:

The general objectives of this course are as follows:

- To provide knowledge about the fundamentals of server-side developmentusing appropriate programming language.
- To enable students to develop web applications using the framework and manage databases in web applications.
- To introduce students to the basics of Content Management Systems (CMS).
- To develop skills in creating and consuming web services and APIs.
- To equip students with the knowledge to implement security best practices in web applications.
- To prepare students for the deployment of web applications to production environments.

3. Methods of Instructions:

This course can be delivered through a combination of lectures, hands-on coding exercises, and practical projects. Students will engage in interactive sessions that involve building real-world web applications and practicing new skills through lecturer and assignments. Each unit will include both theoretical explanations and practical tasks to reinforce learning and ensure students can apply their knowledge effectively.

4. Course Contents:

Specific Objectives	**Contents**
Unit 1: Introduction to Server-Side Development	**4 hours**
• Describe web technologies role of client-side verses server-side. • Explain HTTP request and response message format and different web servers. • Explore server architectures and backend programming languages. • Implement MVC design in web development.	1.1. Overview of Web-I 1.2. HTTP request message, HTTP response Message 1.3. Differences between client-side and server-side technologies 1.4. Web Clients 1.5. Web Server 1.5.1. Role of the server in a web application 1.5.2. Overview of web servers - Apache - Nginx - IIS 1.6. Server-side architecture 1.6.1. Monolithic 1.6.2. Micro-services 1.7. Introduction to backend programming languages (Python, PHP, Java) 1.8. Understanding MVC (Model-View-Controller) architecture

Unit 2: Server-Side Programming with Python	**8 hours**
• Explain about Python data types, functions, and control structures. • Explore about lists, dictionaries, files, and understand OOP concepts. • Implement CSV and JSON files	2.1 Setting up the Python environment 2.2 Introduction to Python syntax and basic operations 2.3 Data Types and Variables 2.3.1. int, float, str, bool, list, dict, tuple, set 2.3.2. Variable assignment and naming conventions 2.3.3. Type conversion 2.4 Control Structures 2.4.1 Conditional statements: if, elif, else 2.4.2 Loops: for and while 2.4.3 List comprehensions 2.5 Introduction to functions 2.5.1 Defining functions, parameters, return values 2.5.2 Scope and global variables 2.6 Working with Python Data Structures (1 hour) 2.6.1 Lists 2.6.2 Creating, accessing, and modifying lists List methods: append(), remove(), pop(), etc. 2.6.3 Dictionaries: Key-value pairs, adding and accessing items Dictionary methods: get(), keys(), values() 2.6.4 Tuples and Sets: Creating and using tuples and sets 2.6.5 Differences between lists, tuples, and sets 2.7 File Handling in Python 2.7.1 Introduction to File Operations 2.7.2 Opening and closing files 2.7.3 File modes 2.7.4 Reading from Files Reading entire files, line-by-line reading 2.7.5 Using with statement for file operations 2.7.6 Writing to Files: Writing data to files, Appending data to existing files 2.8 Working with CSV and JSON Files 2.8.1 Reading and writing CSV files using csv module 2.8.2 Handling JSON files using json module 2.9 OOP Concepts
Unit 3: Working with Web Framework	**8 hours**
• Describe the importance of framework in web development • Implement HTTP requests, form submissions, cookies, and sessions • Explore Flask features like file uploads and sending emails	3.1 Framework overview: 3.2 Introduction to Flask Framework 3.3 Setting up a Flask environment 3.4 Flask basics: Routes, Views, and Templates 3.5 Handling HTTP requests and responses 3.6 Building Basic Web Functionality 3.6.1 Implementing form submission and input validation 3.6.2 Understanding and managing cookies and sessions 3.7 Flask redirect, Message flashing , File upload 3.8 Sending Email
Unit 4: Database Integration	**8 hours**

• Implement web application with database • Implement basic CRUD operations and manage database transections.	4.1 Introduction to Databases 4.2 Overview of relational (SQL) vs. NoSQL databases 4.3 Database Connectivity with Flask 4.3.1 Connecting Flask applications to databases using SQL Alchemy 4.4 Performing CRUD operations using HTML template 4.5 Handling database connections and transactions
Unit 5: Introduction to Content Management Systems	**3 hours**
• Explain the knowledge about CMS and its importance in web. • Describe the knowledge of the popular frameworks.	5.1 Overview of popular CMS platforms (WordPress, Drupal, Joomla) 5.2 Advantages of using CMS in web development 5.3 Use cases and scenarios where CMS is beneficial 5.4 Brief about Web-programming frameworks: php, java, python frameworks.
Unit 6: Web Services and APIs	**5 hours**
• Describe SOA,SOAP, API , JSON and XML • Implement REST API.	6.1 Introduction to Web Services 6.2 Brief about service-oriented architecture 6.3 SOAP 6.4 RESTful web services 6.6 RESTful APIs vs. SOAP 6.6 JSON and XML data formats 6.7 Building and Consuming APIs 6.7.1 Creating simple RESTful APIs using Flask
Unit 7: Security in Web Applications	**8 hours**
• Analyse the security basics and common vulnerabilities. • Explore security principles and best practices. • Implement protection measures for your web apps.	7.1 Web application security fundamentals 7.1.1 Foundations of security 7.1.2 Vulnerabilities 7.1.3 Threats 7.1.4 Attacks, 7.2 Security Principles: 7.2.1 Least Privilege 7.2.2 Defense in Depth 7.2.3 Fail Securely 7.3 Common web vulnerabilities 7.3.1 SQL Injection, 7.3.2 XSS 7.3.3 CSRF 7.3.4 DoS 7.3.5 Live examples and demonstrations of vulnerabilities 7.4 Security best practices for web development 7.4.1 Input Validation and Output Encoding 7.4.2 Secure Authentication and Authorization Multi-Factor Authentication (MFA),Role-Based Access Control (RBAC), Secure Session Handling, Session Fixation and Hijacking Prevention, Error Handling and Logging, Secure File Uploads 7.4.3 SSL/TLS and HTTPS 7.5 Firewalls, proxies, and VPNs 7.6 Host and Network Security threats and countermeasures 7.7 Application Security threats and countermeasures 7.8 Design guidelines for secure web applications
Unit 8: Web Application Deployment	**4 hours**
• Explore the deployment and deployment platforms	8.1 Introduction to Deployment 8.2 Deployment platforms

• Implement SSL certificates to web app or website	8.3 Basic differences between development and production environments. 8.4 Setting up environment variables and basic production settings. 8.5 Introduction to using Gunicorn to serve Flask apps. 8.6 Setting up basic HTTPS using a free certificate (e.g., Let's Encrypt). 8.7 Brief introduction to monitoring and logging. 8.8 Importance of security in deployment.

5. Practical Works:

Practical works in this course covers all the tools and techniques for development and operation of technological enhancements for fulfilling the market current demands. Students will gain hands-on experience in building, securing, and deploying web applications.

(Practical works are outline as follows but not limited.)

1. Create a basic Flask application with a route that returns "Hello, World!" when accessed via the browser.
2. Develop a web form using HTML and Flask to capture user input (e.g., name, email) and display the input on another page.
3. Installation and setting MySQL database server.
4. Write a Flask application that connects to a MySQL database using MySQL-connector or SQL Alchemy. Create a table called students with columns id, name, age, and grade. Insert records and fetch data from the database to display on a webpage.
5. Implement HTML Dropdown to display dynamic options, Dynamic HTML tables, product list and product details using the flask.
6. Create a simple MySQL database and connect it to a Flask application. Develop a form that inserts user data into the database.
7. Create a Flask app that allows users to add, update, and delete records from a database using a web interface.
8. Use Bootstrap with Flask to style the CRUD application developed in above task. Create a form with proper layout and design. Use Bootstrap classes for buttons, tables, and forms.
9. Implement input validation for the form created in the previous exercise.
10. Develop a login page using Flask that includes session management and secure password storage using hashing
11. Develop a REST API using Flask that performs CRUD operations on a student's table in the MySQL database.
12. Implement API endpoints for creating, retrieving, updating, and deleting student records.
13. Use Python's requests module to fetch data from an open API (e.g., weather or currency exchange API) and display the results.

6. Evaluation System and Students' Responsibilities:

6.1 Evaluation System:

In addition to the formal exam(s) conducted by the Office of the Controller of Examination of Pokhara University, the internal evaluation of a student may consist of class attendance, class participation, quizzes, assignments, presentations, written exams, etc. The tabular presentation of the evaluation system is as follows.

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		**30**	**Semester End examination**	**50**
Attendance / Class Participation	10%			
Assignments	20%			
Project Work/Presentations	10%			
Term Exam	60%			
Practical		**20**		
Attendance and Lab Participation	10%			
Lab Report	20%			
Lab Examination	40%			
Viva Examination	30%			
Total Internal Marks		**50**		
Full marks=50+50				

6.2 Students' Responsibilities:

To be eligible for the Semester End Examinations, each student must secure at least 45% marks in the internal evaluation with 80% attendance in the class to appear in the Semester End Examination. Failing to obtain such score will be given NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Term examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during the period. If a student fails to attend a formal exam, quiz, test, etc. there won't be any provision for a re-exam.

7. Prescribed Books and References:

1. Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python (2nd ed.). O'Reilly Media.
2. Sweigart, A. (2020). The Big Book of Small Python Projects: 81 Easy Practice Programs. No Starch Press.
3. Beazley, D. (2020). *Python distilled*. Addison-Wesley.

Pokhara University
Faculty of Science and Technology

Course Code: CMP 323 (2 Credits)		Full Marks: 100
Course Title: Project II (2-1-3)		Pass Marks: 45
Nature of the Course: Theory/Practical		Total Lectures: 48 hours
Level: Bachelor	Year: II / Semester: IV	Program: Bachelor of Computer Application

1. Course Description:
Project-II is an immersive, hands-on course designed for BCA fourth-semester students to translate their classroom knowledge into a tangible, real-world application. The project should focus on the development of a web or desktop application. This course allows students to demonstrate their skills in areas such as user interface design, backend development, and database management. Building a project in web platform or desktop sharpens students' abilities in advanced software development and effective project management.

2. General Objectives:

1) To empower students to apply the knowledge and skills gained in courses like Java, Web Technologies, DBMS to create a practical software application.
2) To help students develop essential project management, teamwork, and communication skills that are crucial for their future careers.
3) To encourage students to think critically and solve problems creatively as they work on their projects.
4) To enable students to produce a functional, user-friendly, and well-documented software application that meets specific needs.

3. Procedure:
The Project II is focused to implement the object oriented system concept in programming using Java, Web Technology, DBMS. A group of maximum four students can be formed to work in their project II. Students are themselves responsible to form the group and select project on any domain of common interest. Project supervisor(s) shall be assigned for guidance, tutoring and mentoring of the project activities. The project shall be evaluated by evaluation panel that includes internal evaluator(s) and/or supervisor and external examiner(s) appointed by the respective department. Appointment of External Examiner is mandatory for final defense of the project. Each group is required to complete all project work phases and develop software product that is SMART (Specific, Measurable, Achievable, Relevant, and Time-bound.

4. Project Phases:

The entire project work shall be divided into three phases and evaluation shall be done accordingly:

4.1 Phase I – Proposal Submission and Defense
In Phase I, students are required to form a team comprising of maximum 4 team members and draft a conceptual framework for their project work in the form of a Proposal. The project team must submit and present the proposal in presence of Examiner/Supervisor in a formal presentation and must be approved.

4.2 Phase II – Mid Term Defense:
The project team needs to present the work progress to Examiner(s)/Supervisor and demonstrate as required. Till the mid-term defense, students must have finished the design phase including the overall system/architectural design and validation scheme. The Mid-Term defense can be scheduled after 4 weeks of project proposal defense.

4.3 Phase III – Final Submission and Defense:
Before the semester ends, students must submit and defend the project final work (oral presentation and demonstration) in presence of the evaluation panel inclusive of external examiner. The project work should be presented in complete format that include project requirement analysis, design, coding and testing. Students are required to submit the final report 1 week prior to the final defense. Students must submit the hardcopy/softcopy of the project in the standard format prescribed by respective department of the college.

6. Documentation Structure:

The documentation structure defines the components that must be included in the proposal and final report of the Project II. It includes the following components:

Proposal Format:

1. Introduction
2. Problem statement
3. Objectives
4. System requirement
   - 4.1 Functional
   - 4.2 Non functional
   - 4.3 Requirement prioritization table
5. Design
6. Development Cost estimation
7. Project Schedule
8. Conclusion
9. References

Report format:

Title Page
Student Declaration
Supervisor Acceptance
Approval Certificate
Acknowledgement
Abstract
List of Abbreviations
Table of Figures
List of Tables
Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Background
1.2 Objectives
1.3 Purpose, Scope, and Applicability
   - 1.3.1 Purpose
   - 1.3.2 Scope and Limitation
   - 1.3.3 Applicability

1.4 Achievements
1.5 Organization of Report

CHAPTER 2: SURVEY OF TECHNOLOGIES

Review of the similar/relevant projects

CHAPTER 3: REQUIREMENTS AND ANALYSIS

3.1 Problem definition
3.2 Requirements specification
3.3 Planning and scheduling
3.4 System requirements
3.5 Preliminary product description

CHAPTER 4: DESIGN

4.1 Introduction
4.2 System Design
4.3 Database design
4.4 Interface Design

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation Approaches
5.2 Coding Details and Code Efficiency