THIRD SEMESTER
(Detailed Syllabus)

'ARTIFICIAL INTELLIGENCE
&
MACHINE LEARNING'

DISCRETE MATHEMATICS

Course Code:	433001
Course Title	Discrete Mathematics
No. of Credits	4 (TH:4,T:0,P:0)

Course Outcomes: At the end of the course, the student will be able to:

1. Understand fundamental concepts of discrete mathematics.
2. Apply discrete mathematics principles to solve practical problems in Artificial Intelligence and Machine Learning.
3. Analyse basic algorithms and data structures used in relevant field.
4. Identify graph theory applications in Artificial Intelligence and Machine Learning analysis.
5. Demonstrate basic knowledge of set theory, logic, and combinatorics relevant to Artificial Intelligence and Machine Learning.

COURSE CONTENTS

Unit 1: Sets and Logic

- Introduction to sets and set operations
- Logic and truth tables
- Propositional logic and logical operators

Unit 2: Combinatorics

- Counting principles: Permutations and combinations
- Basic probability concepts and applications

Unit 3: Graph Theory

- Introduction to graphs and their components
- Graph representations and basic graph algorithms
- Applications of graph theory in AI&ML (over view only)

Unit 4: Number Theory

- Divisibility and prime numbers
- Modular arithmetic and its applications in cryptography (over view only)

Unit 5: Relations and Functions

- Introduction to relations and properties
- Functions and their properties
- Applications of relations and functions in AI&ML (over view only)

Suggested Books:

1. "Discrete Mathematics and its Applications" by Kenneth H. Rosen
2. "Discrete Mathematics: Elementary and Beyond" by L. Lovász, J. Pelikán, and K. Vesztergombi
3. "A Path to Combinatorics for Undergraduates" by Titu Andreescu and Zuming Feng
4. "Introduction to Graph Theory" by Richard J. Trudeau
5. "Introduction to the Theory of Computation" by Michael Sipser

*****

COMPUTER SYSTEM ORGANIZATION

Course Code:	433002
Course Title	Computer System Organization
No. of Credits	4 (TH:4,T:0,P:0)

COURSE OUTCOMES: After completion of this course, student will be able to:

1. Have a good understanding of functioning of computer system as such and its various sub- components. Student will be able to understand computing requirement for a specific purpose, analyze performance bottlenecks of the computing device and choose appropriate computing device for a given use case.
2. Understand a computer system that has hardware and software components, which controls and makes them useful.
3. Understand the fixed and floating point number representation in computer.
4. Understand how arithmetic operation will be performed in computer system.
5. Acquire knowledge on Cache and virtual memory.
6. Understand Interrupt and DMA access.
7. Acquire knowledge on RISC and CISC architecture.
8. Understand how pipelining and parallel processing improves the performance of computer system.

COURSE CONTENTS

UNIT - 1: **Basics of Computer system**

1.1 Structure of Computers: Computer Functional units, Von-Neumann architecture and its bottleneck, Bus structures, Basic Operational Concepts, Data representation (Fixed and Floating point), Error detecting codes.

1.2 Register Transfer and Micro Operations: Register transfer, Bus and memory transfers, Arithmetic micro-operations, Logic micro-operations, Shift micro-operations, and Arithmetic logic shift unit.

UNIT - 2 : **Introduction to Microprocessor Architecture**

2.1 Difference between Computer Architecture and Organization, Instruction Set Architecture design principles from programmer's perspective. One example of microprocessor.

UNIT - 3 : Assembly Language Programming

3.1 Simple programs, Assembly language programs involving logical, branch and call instructions, sorting, evaluation of arithmetic expressions, string manipulation, assembler directives, procedures and macros.

UNIT - 4 : Memory and Digital Interfacing

4.1 Addressing and address decoding, interfacing RAM, ROM, EPROM, programmable peripheral interface, various modes of operation and interfacing to processor, interfacing keyboard, displays, etc.

4.2 Memory and I/O: Memory Hierarchy model and comparison on cost, speed and size; Cache memory, Locality of Reference, Concept of virtual memory technique, Overview of different methods of IO access mechanism; Programmed IO or Status check IO, Interrupt Classification, DMA data transfer, IO processor; Overview of DMA transfer, Programmable Interrupt Controller, Programmable DMA Controller.

UNIT - 5 : Control Unit and Parallel Processing

5.1 Micro Programmed Control: Control memory, Address sequencing, and design of control unit.

5.2 Overview of Basic Computer Arithmetic operations.

Text Books:

1. Computer System Architecture, M. Moris Mano, Pearson/PHI, India.
2. Computer Organization, Carl Hamacher, Zvonks Vranesic, SafeaZaky, McGraw-Hill
3. Computer Organization and Architecture, Stallings, Pearson

Reference Books:

1. Microprocessors Interface, Douglas V.Hall, Tata McGraw-Hill.
2. Advanced Microprocessors and Peripherals- Architecture, Programming and interfacing, A.K.Ray, K. M. Bhurchandi, Tata McGraw-Hill, New Delhi, India.
3. Computer Organization and Design: A Hardwar/ Software Interface (MIPS Edition) by Patterson and Hennessy

*****

DATA STRUCTURE USING 'C'

Course Code:	433003
Course Title	Data Structure Using 'C'
No. of Credits	6 (TH:4,T:0,P:4)

COURSE OUTCOMES : At the end of the course, the student will be able to:

1. Understand the fundamental concepts of data structures and their importance in programming.
2. Gain proficiency in implementing and using various data structures such as arrays, linked lists, stacks, queues, trees, and graphs using the C programming language.
3. Develop problem-solving skills by applying appropriate data structures and algorithms to solve real-world programming challenges.
4. Acquire knowledge of different sorting and searching techniques and their implementation using C.
5. Learn the concept of recursion and its applications in solving problems related to data structures.
6. Develop proficiency in using dynamic memory allocation and pointers to efficiently manage and manipulate data structures.

COURSE CONTENTS

1. Fundamental Notations

1.1 Problem solving concept, top down and bottom up design, structured programming

1.2 Concept of data types, variables and constants

1.3 Concept of pointer variables and constants

1.4 Introduction to data Structure (Linear, Non Linear, Primitive, Non Primitive)

1.5 Concepts of Data Structure(Array, Linked List, Stack, Queue, Trees, graphs)

2. Arrays

2.1 Concept of Arrays

2.2 Single dimensional array

2.3 Two dimensional array: Representation of Two dimensional Array (Base Address, LB, UB)

2.4 Operations on arrays with Algorithms (searching, traversing, inserting, deleting)

3. Linked

3.1 Introduction to linked list and double linked list

3.2 Representation of linked lists in Memory, Comparison between Linked List and Array

3.3 Traversing a linked list

3.4 Searching linked list

3.5 Insertion and deletion into linked list (At first Node, Specified Position, Last node

3.6 Application of linked lists

4. Stacks, Queues and Recursion

4.1 Introduction to stacks

4.2 Representation of stacks with array and Linked List

4.3 Implementation of stacks

4.4 Application of stacks

4.5 Recursion: Concept and Comparison between recursion and Iteration

4.6 Introduction to queues

4.7 Implementation of queues

5. Trees: Introduction

5.1 Concept of binary trees (complete, extended binary tree)

5.2 Concept of representation of Binary Tree

5.3 Concept of balanced Binary Tree

5.4 Traversing binary trees (pre order, post order & in order)

5.5 Searching, inserting and deleting in binary search trees

6. Sorting and Searching

6.1 Introduction

6.2 Search algorithm (Linear and Binary)

6.3 Concept of sorting

6.4 Sorting algorithms (Overview only)

PRACTICAL OUTCOMES : At the end of the course, the student will be able to:

1. Gain hands - on experience in implementing various data structures using the C programming language.
2. Develop proficiency in using C language constructs and functions to create and manipulate data structures such as arrays, linked lists, stacks, queues, trees, and graphs.
3. Acquire practical knowledge of implementing algorithms for sorting, searching, and other operations on different data structures.
4. Enhance problem-solving skills by applying appropriate data structures and algorithms to solve programming challenges in a laboratory setting.
5. Develop the ability to analyze and evaluate the efficiency and performance of different data structures and algorithms through practical experiments and benchmarking.
6. Improve teamwork and collaboration skills by working on group projects and assignments that involve implementing and testing data structures using C programming.

List of Practicals :-

Write Programs in 'C' language to implement:-

1. The addition of two matrices using functions.
2. Inserting and deleting elements in array.
3. Push and pop operation in stack.
4. Conversion from in-fix notation.
5. The factorial of a given number using recursion.
6. Insertion and Deletion of elements in queue using pointers.
7. Insertion and Deletion of elements in circular queue using pointers.
8. Insertion and Deletion of elements in linked list.
9. Insertion and Deletion of elements in doubly linked list.
10. The linear search procedures to search an element in given list.
11. The binary search procedures to search an element in a given list.

References /Suggested Learning Resources:

1. Digital principles & Applications Albert Paul Malvino & Donald P. Leach McGraw Hill Education; Eighth edition ISBN: 978-9339203405
2. Digital Electronics Roger L. Tokheim Macmillan McGraw-Hill Education (ISE Editions); International 2 Revised edition ISBN: 978-0071167963
3. Digital Electronics – an introduction to theory and practice William H. Gothmann Prentice Hall India Learning Private Limited; 2 edition ISBN: 978-8120303485
4. Fundamentals of Logic Design Charles H. Roth Jr. Jaicoh Publishing House; First edition ISBN: 978-8172247744
5. Digital Electronics R. Anand Khanna Publications, New Delhi (Edition 2018) ISBN: 978-93-82609445

*****

‘C ‘ PROGRAMMING

Course Code:	433004
Course Title	‘C ‘ Programming
No. of Credits	6 (TH:4,T:0,P:4)

COURSE OUTCOMES: At the end of the course, the student will be able to:

1. Develop algorithms and flowcharts to solve programming problems and understand the steps involved in program development.
2. Demonstrate proficiency in writing and executing C programs, including understanding program structure, I/O statements, variables, data types, and storage classes.
3. Apply control structures such as decision-making statements (IF-ELSE, nested IF, and switch), loops (while, do-while, for), and break/continue statements to control program flow.
4. Understand the concept of functions, including global and local variables, function declaration and calling, different types of functions, and parameter passing.
5. Manipulate arrays and strings effectively, including array declaration, accessing array elements, multidimensional arrays, and using string-related functions.
6. Demonstrate understanding of pointers, including static and dynamic memory allocation, working with addresses and pointers, and utilizing pointers to arrays and structures.

DETAILED CONTENTS

1. Algorithm and Programming Development

Steps in development of a program, algorithm development, concept of flowcharts, programming & use of programming, various techniques of program-ming, Structured Programming, Pre-processors, Debugging, Compiling.

2. Program Structure

Structure of C program, Writing and executing the first C program, Translator: Assembler, Interpreter, Compiler, I/O statement, assign statement, Keywords, constants, variables and data types, storage classes, operators and expressions, Unformatted and Formatted IOS, Data Type Casting

3. Control Structures

Introduction, decision making with IF – statement, IF – Else and Nested IF, Ladder, if-else, Loop: While, do-while, for, Break, Continue, goto and switch statements.

4. Functions

Introduction to functions, Global and Local Variables, Function Declaration, Function Call and Return, Types of Functions, Standard functions, Parameters and Parameter Passing, Call-by value/ reference, recursive function, function with array, function with string.

5. Arrays and Strings

Introduction to Arrays, Array Declaration, Length of array, Manipulating array elements, Single & Multidimensional Array, Arrays of characters, Passing an array to function, Introduction of Strings, String declaration and definition, Overview of String Related function.

6. Pointers

Introduction to pointers, Static and dynamic memory allocation, Address operator and pointers, Declaring and initializing pointers, Single pointer, Pointers to an array.

7. Structures and Unions

Declaration of structures, Accessing structure members, Structure initialization, array of structure variable, Pointer to a structures, Union, Declaration of Union.

8. File Handling

Basics of File Handling, opening and closing of File, reading and writing character from a file.

PRACTICAL OUTCOMES: After undergoing the subject, the students will be able to:

1. Identify the problem and formulate an algorithm for it.
2. Use pointer in an array and structure.
3. Use structures and union for data handling.
4. Install C software on the PC and debug the programme.
5. Explain & execute member functions of C in the programme
6. Describe and implement array concept in C programme
7. Expose File System using File Handling.

List of Practicals :-

1. Programming exercises on executing & editing a C program.
2. Programming exercises on defining variables and assigning values to variables.
3. Programming exercises on arithmetic, logical and relational operators.
4. Programming exercises on arithmetic expressions and their evaluation.
5. Programming exercises on formatting input/output using printf and scanf & their return type values.
6. Programming exercises using if statement.
7. Programming exercises using if – Else.
8. Programming exercises on switch statement.
9. Programming exercises on while and do– while statement.
10. Programming exercises on for – statement.
11. Simple programs using functions and recursive function.
12. Programming Exercise on array.
13. Simple programs using pointers.
14. Simple programs using structures.
15. Simple programs for File Handling

References :-

1. Problem Solving and Programming in C, R.S. Salaria, Khanna Publishing House
2. C Programming Absolute Beginner's Guide, Dean Miller and Greg perry
3. The C Programming Language, Kernighan and Ritchie, Prentice Hall of India

*****

WEB TECHNOLOGIES

Course Code:	433005
Course Title	Web Technologies
No. of Credits	6 (TH:4,T:0,P:4)

COURSE OUTCOMES: At the end of the course, the student will be able to:

1. Understand the fundamental concepts and principles of web technologies, including client-server architecture, HTTP, and web standards.
2. Design and develop interactive and user-friendly web pages using HTML, CSS, and JavaScript.
3. Implement responsive web design techniques to create websites that adapt to different screen sizes and devices.
4. Utilize server-side scripting languages (such as PHP or ASP.NET) to create dynamic web pages and handle form submissions.
5. Integrate databases into web applications and perform basic database operations using SQL.
6. Apply web security principles and best practices to protect against common vulnerabilities and threats.

COURSE CONTENTS

UNIT - 1: Introduction to WWW

1.1 Protocols and programs

1.1.1 Secure connections,

1.1.2 Application and development tools

1.2 The web browser

1.3 What is server

1.4 Setting up UNIX and LINUX web servers

1.5 Logging users

1.6.Dynamic IP Web Design

1.6.1 Web site design principles

1.6.2 Planning the site and navigation

UNIT - 2 : Web Systems Architecture (Overview Only)

2.1 Architecture of Web based systems

2.2 Building blocks of fast and scalable data access Concepts

2.3 Web Application architecture (WAA)

UNIT - 3: Javascript

3.1 Client-side scripting

3.2 Basic Concept of JavaScript

3.3 Variables, Functions, conditions, Loops and repetition

UNIT 4: Advance Scripting

4.1.JavaScript and objects

4.1.1 JavaScript own objects

4.1.2 DOM and web browser environments, forms and validations

4.2 DHTML

4.3 Introduction to Ajax

4.4 Introduction to XML

4.5 Introduction to Web Services

UNIT - 5 : PHP

5.1 Server- side scripting

- 5.1.1 Arrays
- 5.1.2 Function and forms
- 5.1.3 Advance php

5.2 Databases

- 5.2.1 Basic command with php examples
- 5.2.2 Connection to server, creating database
- 5.2.3 Selecting and listing database
- 5.2.4 Listing table- names creating a table
- 5.2.5 Inserting data

PRACTICAL OUTCOMES: By the end of the web technology lab, the student will be able to:

1. Demonstrate proficiency in developing server-client programs using various programming languages & protocols.
2. Create dynamic web pages using HTML and enhance interactivity using JavaScript.
3. Apply advanced techniques of CSS to design visually appealing and user-friendly web applications.
4. Develop PHP programs by applying the basics of PHP programming language.
5. Build web applications using PHP, including form handling, session management, & integration with databases.
6. Connect PHP with MySQL database and develop secure tiered web applications.
7. Design and develop a fully functional web service application by integrating all the technologies learned throughout the course.

List of Practicals:

1. Coding Server Client Programs
   - Practice developing server-client programs using various programming languages and protocols.
2. Developing Web Applications using HTML and JavaScript
   - Learn and practice creating dynamic web pages using HTML for structure and JavaScript for interactivity.
3. Developing Advanced Web Application Programs using CSS

- Explore advanced techniques of Cascading Style Sheets (CSS) to design and enhance web applications.

4. Practicing PHP: Basics
   - Gain hands-on experience in PHP programming language, covering its syntax, variables, data types, and control structures.
5. Practicing PHP: Web Application Development
   - Develop web applications using PHP, focusing on form handling, session management, and integrating databases.
6. Practicing PHP: MySQL - Tiered Applications
   - Learn how to connect PHP with MySQL database and build tiered web applications with secure data handling.
7. Developing a Fully Functional Web Service Application
   - Apply all the technologies learned throughout the course to develop a comprehensive and functional web service application.

References:

1. "Web Technologies--A Computer Science Perspective", Jeffrey C. Jackson,
2. "Internet & World Wide Web How To Program", Deitel, Deitel, Goldberg, Pearson Education
3. "Web programming- Building Internet Application", Chris Bales
4. "Web Applications: Concepts and Real World Design", Knuckles

*****

E-GOVERNANCE

Course Code:	433006
Course Title	E-Governance
No. of Credits	6 (TH:4,T:0,P:4)

COURSE OUTCOMES: After Completion of the course the student will be able to:

1. Develop an understanding of the emerging trends in ICT for development and their application in the context of e-Governance.
2. Acquire knowledge of the design and implementation of e-Government projects, along with the lifecycle involved.
3. Recognize the need for Government Process Re-engineering (GPR) and its role in improving governance practices.
4. Understand the National e-Governance Plan (NeGP) for India & its impact on transforming governance processes.
5. Gain insights into the architecture and models of e-Governance, including Public Private Partnership (PPP) approaches.
6. Develop skills in evaluating e-Governance projects and identifying critical success factors, while addressing challenges such as corruption, resistance to change, e-Security, and Cyber laws.

COURSE CONTENTS

UNIT - 1

Exposure to emerging trends in ICT for development; Understanding of design and implementation of e-Government projects, e-governance lifecycle.

UNIT - 2

Need for Government Process Re-engineering (GPR); Features of National e-Governance Plan (NeGP) for India; SMART Governments & Thumb Rules.

UNIT - 3

Architecture and models of e-Governance, including Public Private Partnership (PPP); Need for Innovation and Change Management in e-Governance; Critical Success Factors; Major issue including corruption, resistance for change, e-Security and Cyber laws.

UNIT - 4

Focusing on Indian initiatives and their impact on citizens; Sharing of case studies to highlight best practices in managing e-Governance projects in Indian context. Visits to local e-governance sites (CSC, eSeva, Post Office, Passport Seva Kendra, etc) as part of Tutorials.

UNIT - 5

Mini Projects by students in groups – primarily evaluation of various e-governance projects.

PRACTICAL OUTCOMES: At the end of the course, the student will be able to:

1. Gain practical skills in the design and implementation of e-Government projects using relevant software tools.
2. Analyze and evaluate the impact of successful e-Governance case studies on citizens.
3. Apply government process re-engineering techniques and propose innovative solutions using ICT.
4. Demonstrate knowledge of e-Security measures and compliance with cyber laws in e-Governance systems.
5. Develop critical evaluation skills by assessing the effectiveness and efficiency of different e-Governance initiatives.
6. Gain practical exposure and understanding of e-Governance implementation through site visits to local e-Governance centers.

List of Practicals:

1. Conducting a hands-on session on the design and implementation of e-Government projects, where students can work with relevant software tools and technologies used in e-Governance.
2. Exploring case studies of successful e-Governance projects in India and analyzing their impact on citizens, including factors such as user experience, accessibility, and effectiveness.
3. Simulating a government process re-engineering (GPR) exercise, where students can identify areas for

improvement in existing government processes and propose innovative solutions using ICT.

4. Analyzing and assessing the security aspects of e-Governance systems, including conducting vulnerability assessments and proposing measures to enhance e-Security and comply with cyber laws.
5. Conducting group mini projects where students can evaluate various e-Governance projects, studying their effectiveness, efficiency, user satisfaction, and impact on society. Students can present their findings and recommendations based on their evaluation.
6. Organizing visits to local e-Governance sites, such as Common Service Centers (CSCs), eSeva centers, Post Offices, Passport Seva Kendras, or other relevant government service delivery centers. This allows students to observe and understand the practical implementation of e-Governance initiatives.

Reference Books:

1. Managing Transformation –Objectives to Outcomes. J Satyanarayana, Prentice Hall India
2. The State, IT and Development. Kenneth Kenniston, RK Bagga and Rohit Raj Mathur, Sage Publications India Pvt Ltd.
3. e-Government -The Science of the Possible. J. Satyanarayana, Prentice Hall, India
4. http://www.csi-sigegov.org/publications.php
5. https://negd.gov.in
6. https://www.nisg.org/case-studies-on-e-governance-in-

DATA COMMUNICATION LAB

Course Code:	433007
Course Title	Data Communication Lab
No. of Credits	2 (TH:0,T:0,P:4)

PRACTICAL OUTCOMES: By the end of this course, the student will be able to:

1. Understand data communication fundamentals and its significance in modern networking.
2. Identify and differentiate between LAN, WAN, and MAN networks.
3. Demonstrate practical knowledge of guided and unguided transmission media.
4. Configure network devices and implement various topologies while troubleshooting issues.
5. Comprehend TCP/IP and OSI models, implement common protocols, design wired and wireless networks, and grasp cellular network generations and evolution.

Practical Exercises:

1. a. Setting up a basic LAN using Ethernet cables and switches.
   b. Configuring network devices (routers/switches) to establish communication between different LANs.
2. a. Testing the performance of different guided media (twisted pair, coaxial cable, fibre-optic cable) for data transmission.
   b. Measuring signal attenuation and bandwidth of different transmission media.
3. a. Configuring routers and switches to implement different network topologies (bus, star, ring, mesh) in a lab environment.
   b. Analysing the advantages and disadvantages of each network topology in terms of scalability and fault tolerance.
4. a. Capturing and analysing network packets using Wireshark to understand TCP/IP and OSI model layers.
   b. Configuring and testing common data communication protocols (TCP, UDP, IP, HTTP, FTP) in a network setup.
5. a. configuring and securing wireless LANs using IEEE 802.11 standards (Wi-Fi).
   b. Hands-on experience with different generations of cellular networks and comparing their performance and features.

References:

1. "Data Communications and Networking" by Behrouz A. Forouzan
2. "Computer Networking: A Top-Down Approach" by James F. Kurose and Keith W. Ross
3. "TCP/IP Illustrated, Volume 1: The Protocols" by W. Richard Stevens
4. "Network Warrior: Everything You Need to Know That Wasn't on the CCNA Exam" by Gary A. Donahue
5. "Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall
6. "Wireless Communications & Networks" by William Stallings
7. "Data and Computer Communications" by William Stallings

*****

SUMMER INTERNSHIP-I

Course Code:	AS301
Course Title	Summer Internship - I
No. of Credits	2 (TH:0,T:0,P:0)

Summer Internship provides an invaluable opportunity for students pursuing their Diploma in Engineering to gain real-world experience and exposure to various industrial production units and commercial activities related to their field of study. This program aims to bridge the gap between theoretical knowledge and practical application, equipping students with the necessary skills and expertise to thrive in the branch related industry.

At the end of the **Second semester**, students will undertake a minimum 3 to 4-week Summer Internship, scheduled during the semester break following the Second Semester examinations. The respective Heads of Departments (HoDs) and experienced faculty members will guide and assist students in securing suitable training opportunities that align with their specialization. Each student will have a personalized training schedule developed in collaboration with the training providers, ensuring a comprehensive and enriching learning experience.

Before starting their training, students will receive a comprehensive briefing about the organizational setup, product range, manufacturing processes, and significant machinery and materials used in the training organization. This preliminary understanding will enhance their engagement and productivity during the internship.

To ensure a fruitful learning experience, faculty members will supervise students during their training in the industry or field organization. Each faculty member will mentor a small group of 4-5 students, providing personalized attention and guidance. Students will be encouraged to maintain daily reports in their diaries, which will assist them in composing their final training report and presentation.

The evaluation process for the Summer Internship will include both internal and external assessments, as per the study and evaluation scheme of the **Third Semester**. During the viva - voce / presentation examination, students' understanding of materials, industrial processes, practices in the industry, and problem-solving abilities will be assessed. The evaluation will also focus on their application of knowledge and skills in real-life situations.

The components of evaluation will comprise:

(a) Punctuality and regularity: 15%

(b) Initiative in learning new things: 15%

(c) Relationship with peers and colleagues: 10%

(d) Summer Internship report: 25%

(e) Viva-Voce: 35%

The purpose of summer Internship program is to create a transformative experience for students, empowering them to excel in their future careers and make meaningful contributions to the Engineering industry. The collaborative efforts of experienced faculty members and industry partners will ensure that students may gain valuable insights and practical skills during this immersive learning journey.

*****