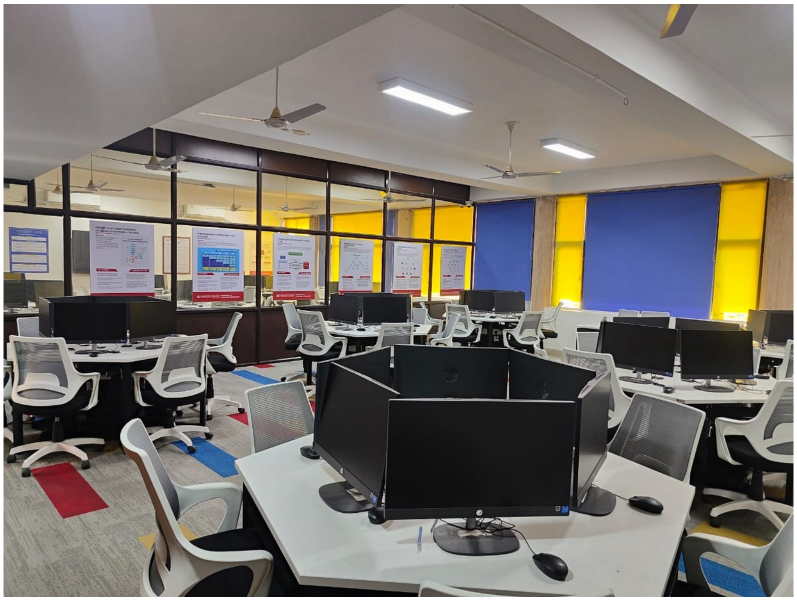
Software Engineering Lab

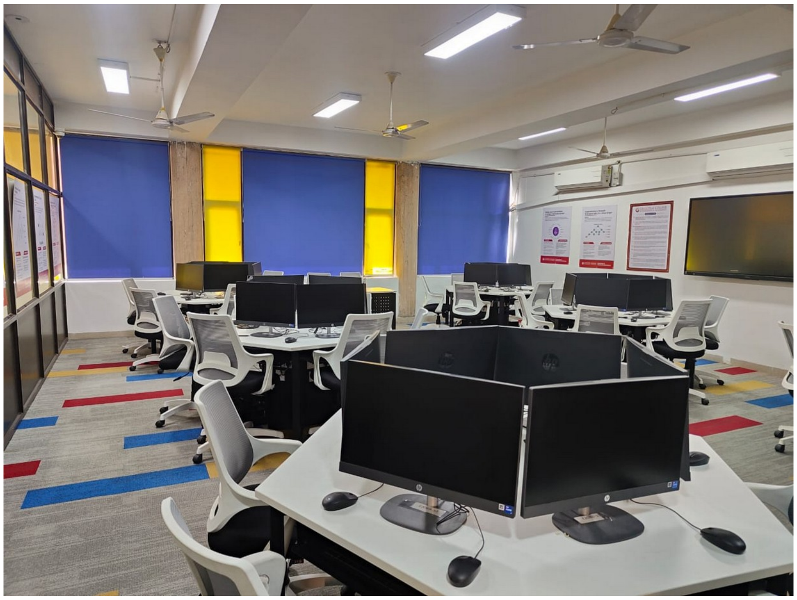
Department of Information Technology

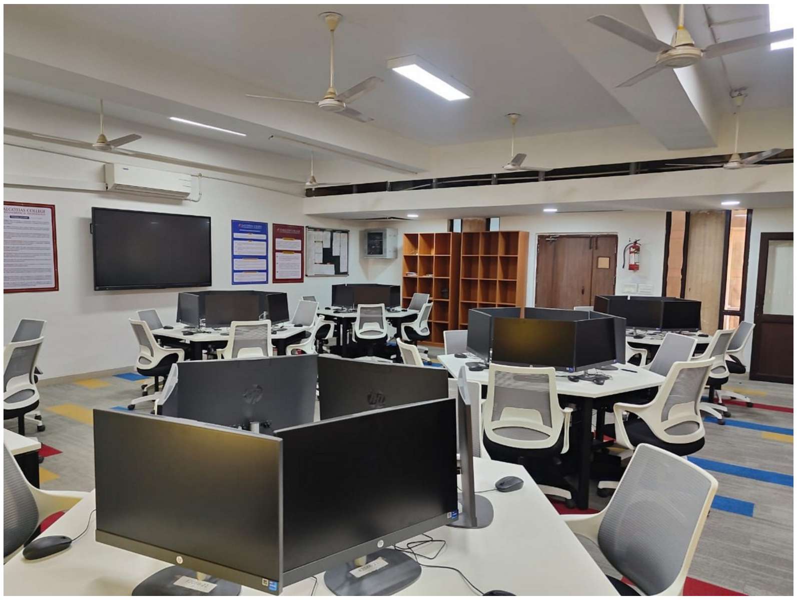
Lab Description:

The Software Engineering Lab offers hands-on experience in applying software development principles and methodologies. Students practice requirements analysis, system design, modeling, testing, and documentation using standard tools and techniques. The lab emphasizes teamwork, project planning, and quality assurance. It helps students understand the complete software development life cycle and industry best practices.

Lab Photos:







Available Software:

S.No.	Software Name
1	Windows 11
2	Ms Office
3	Dev C++
4	Java(JDK 24)
5	MySQL 8.0
6	Cisco Packet Tracer
7	Python 3.12
8	Android Studio
9	Google Chrome

Major Euipments Available in Lab:

S.NO.	Equipments Name	Items
1	Computer System	36
2	Keyboard	36
3	Mouse	36
4	LED Smart Panel	Samsung 75"

Labs Conducted in E 208b:

Even Sem: Software Engineering Lab

Odd Sem: Artificial Intelligence Lab

List of Experiments – Software Engineering Lab

Subject Code: BCS 651 **Branch: IT**
Subject Name: Software Engineering Lab **Sem: VI**

Course Outcomes	Statement (On completion of this course, students will be able to)
CO1	Identify ambiguities, inconsistencies and incompleteness from a requirements specification and state functional and non-functional requirement
CO2	Identify different actors and use cases from a given problem statement and draw use case diagram to associate use cases with different types of relationship
CO3	Draw a class diagram after identifying classes and association among them

List of Experiments

As per AKTU Syllabus

Program No.	Name of Experiment	CO Mapping
1	Prepare a SRS document in line with the IEEE recommended standards.	CO1
2	Draw the use case diagram and specify the role of each of the actors. Also state the precondition, post condition and function of each use case.	CO2
3	Draw the activity diagram	CO2, CO3
4	Identify the classes. Classify them as weak and strong classes and draw the class diagram.	CO3
5	Draw the sequence diagram for any two scenarios	CO2, CO3

6	Draw the collaboration diagram.	CO2, CO3
7	Draw the state chart diagram.	CO2, CO3
8	Draw the component diagram.	CO2, CO3
9	Perform forward engineering in java. (Model to code conversion)	CO3
10	Perform reverse engineering in java. (Code to Model conversion)	CO3
11	Draw the deployment diagram	CO3

Additional Programs (Content Beyond Syllabus):

Exp. No.	**Name of Experiment**	**CO Mapping**
11	Study different software and development processes	CO1
12	Gather understanding on software developer work requirement	CO2
13	Develop a program module for requirement gathering process	CO3
14	Develop a program how to convert activity diagram and DFD using Programming Language	CO3
15	Design and code a program for Calculator using any programming Language	CO3

List of Experiments –Artificial Intelligence Lab

Subject code: BCS 751 **Branch: IT**

Subject Name: AI Lab **Sem:VII**

Course Outcomes	**Statement(On completion of this course, students will be able to)**
BCS 751.1	Understand and apply basic search algorithms and intelligent problem-solving approaches in Python
BCS 751.2	Design and implement logic-based knowledge representations and reasoning using Python and Prolog.

BCS 751.3	Use Natural Language Toolkit (NLTK) for basic NLP tasks like stemming, tagging, and classification.
BCS 751.4	Analyze and implement basic AI game strategies like Minimax and Alpha-Beta pruning
BCS 751.5	Demonstrate AI techniques in real-world tasks such as text processing, planning, or constraint satisfaction.

As per AKTU Syllabus

Exp. No.	Name of Experiment	As per AKTU Syllabus
1	Implement Breadth First Search (BFS) for a given graph or maze.	CO1
2	Implement Depth First Search (DFS) for a tree or graph structure.	CO1
3.	Solve the 8-Puzzle Problem using A* Search Algorithm.	CO2
4	Implement Hill Climbing Algorithm for numerical optimization or pathfinding.	CO2
5	Implement Simulated Annealing Algorithm for constraint-based search problems	CO3
6	Solve Water Jug Problem using state-space search (BFS or DFS).	CO3
7	Write Prolog programs to define family relationships using predicates	CO3
8	Implement 4-Queens Problem in Prolog using backtracking.	CO4
9	Implement Unification Algorithm in Python or Prolog	CO4
10	Implement Forward and Backward Chaining in a rule-based system (manual or code-based).	CO4
11	Demonstrate Resolution in Propositional Logic through a basic example (e.g., proving a theorem).	CO5
12	Remove punctuation and stop words from a paragraph using nltk.	CO5

Additional Programs (Content Beyond Syllabus):

Exp. No.	Name of	CO Mapping

	Experiment	
15	Implement Tic-Tac-Toe game with a basic AI opponent.	CO4
16	Implement Min-Max (Minimax) Algorithm for decision making in turn-based games	CO5