



*Look forward,
learn modern
knowledge, and
do not waste
time in studies
of old subjects of
no values.*



2020-21



M.C.A. Semester-V



CSM-5571: LABORATORY COURSE – V

DEPARTMENT OF COMPUTER SCIENCE

ALIGARH MUSLIM UNIVERSITY ALIGARH

Computer Lab Manual

Important Instructions to the students in the ongoing Covid-19 Scenario

- In the present Covid-19 scenario, all the labs will be conducted **ONLINE** till the further instructions issued by the University.
- Students are instructed to perform their lab exercises/assignments at their own system from their respective residences.
- During this prevailing situation (COVID-19 crises), the mode (policies/rules) of conducting Lecture/Lab classes will be as per the time to time instructions issued by University.
- The students are advised to complete the weekly activities/assignments well in time (i.e., within the same week).
- The students are also advised to maintain the soft copy of the Lab File of their completed activities/assignments in the prescribed format.

!! Stay Home, Stay Safe !!

CREDITS

The following Lab Manuals Up-gradation Committee members updated the Lab Manuals during lockdown and finally approved in the meeting of Lab Manuals Up-gradation Committee held on 07.07.2020 at 11:30 AM through ONLINE mode using Google Meet:

⇒ **Prof. Rafiqul Zaman Khan (Chairperson)**

⇒ Mr. Suhel Mustajab

⇒ Dr. Aasim Zafar

⇒ Dr. Arman Rasool Faridi

⇒ Dr. Faisal Anwer

⇒ Dr. Mohammad Nadeem

The following committee members originally designed the Lab Manual under the supervision of former Chairperson **Prof. Jamshed Siddiqui**:

⇒ Prof. Mohammad Ubaidullah Bokhari

⇒ Dr. Arman Rasool Faridi

⇒ Dr. Faisal Anwer

⇒ Dr. Aasim Zafar (Convener)

Design & Compilation:

⇒ Mr. S. K. Sharma

Revised Edition: **July 2020**

Approved by BoS: **14.07.2020**

**Department of Computer Science
A.M.U., Aligarh, (U.P.), India**

Lab Manual: LABORATORY COURSE –V (CSM-5571)

CONTENTS	PAGE NO.
COURSE DESCRIPTION	2
CONTENT	2
OBJECTIVES	3
OUTCOMES	4
HOW TO DO WELL IN THIS COURSE	5
TEACHING METHODS AND ASSIGNMENTS	7
SUBMISSION OF DELIVERABLES	8
EVALUATION AND GRADING	8
REQUIRED TEXTS/READING/REFERENCES	9
ATTENDANCE POLICY	10
LATE POLICY	10
MILESTONES FOR MINI-PROJECT	10
TOPICAL CALENDAR	14

COURSE TITLE: LABORATORY COURSE-V

COURSE CODE: CSM – 5571

CREDIT: 4

PERIODS PER WEEK: 6

CONTINUOUS ASSESSMENT: 40 Marks

EXAMS: 60 Marks

COURSE DESCRIPTION

Application of classroom knowledge and skills in computer science to solve real-world problems and to develop research and software development skills.

CONTENT

This course consists of the development of a realistic application, representative of a typical real-life software system, under semi-professional working conditions. The students are expected to propose, analyze, design, develop, test and implement a software system. The student will deliver oral presentations, progress reports, and a final report.

Depending on the topic of the project and the chosen software development methodology, which may vary from one year to another, the following themes may be addressed to some extent:

- Software development methodologies, static (products) and dynamic aspects (processes);
- Requirement analysis (goals, use cases), software architectures, architectural styles and patterns, model-driven engineering (MDE);
- Programming techniques, software development environments, refactoring;
- Software validation through unit tests, integration tests, functional and structural tests, and code reviews.
- Project management, planning, resource estimation, reporting.
- Version management by using a version management tool.
- Examples of kinds of systems to be developed are distributed systems, client/server systems, web based systems, secure systems, mobile systems, adaptable systems, optimizations of existing systems or data-intensive systems, etc.

Besides completing a mini project, the students are required to complete subject related Lab Assignments given by respective course teachers. The individual teachers who are teaching the courses with lab component will be responsible for giving assignments, monitoring and evaluating their respective assignments.

OBJECTIVES

- ❑ To help students develop openness to new ideas in computer science, develop the ability to draw reasonable inferences from observations and learn to formulate and solve new computer science problems using analytical and problem-solving skills;
- ❑ To help students develop the ability to synthesize and integrate information and ideas, develop the ability to think creatively, develop the ability to think holistically and develop the ability to distinguish between facts and opinion;
- ❑ To help students acquire the necessary competences to build a real-life software system by completing different software life cycle phases (like, specification, architecture, design, implementation, validation, documentation, etc);
- ❑ To help students develop the ability to work individually and as part of a team, develop a commitment to accurate work, develop management skills, improve speaking and writing skills, improve the ability to follow directions, instructions and plans, and improve the ability to organize and use time effectively;
- ❑ To help students develop a commitment to personal achievement, the ability to work skillfully, informed understanding of the role of science and technology, a lifelong love of learning, and cultivates a sense of responsibility for one's own behavior and improves self-esteem/self-confidence.

OUTCOMES

Upon successful completion of this course students will be able to:

- ❑ Identify project/research problems; understand information and grasp meaning; translate knowledge into new context; use information, methods, concepts, and theories of fundamental topics in computer science in new situations (Knowledge, Comprehension);
- ❑ Apply computer science principles and practices to a real-world problem; demonstrate in-depth knowledge in the area of the project they have undertaken; solve problems using required knowledge and skills; implement and test solutions/algorithms (Application and Evaluation);
- ❑ Identify potential solutions/algorithms for the project problem; see patterns and modularize the problem, recognize hidden meanings and identify components, show proficiency in software engineering principles (Analysis);
- ❑ Apply a software development methodology currently practiced in industry to produce software system in a rigorous and systematic way using different software life cycle phases (specification, architecture, design, implementation, validation, documentation) (Synthesis);
- ❑ Show evidence (group collaboration, regular meetings, email communications, significant knowledge and skills contributions, etc.) of working productively as an individual and in a team on a project that produces a significant software product (Team Work);
- ❑ Show evidence of competency in oral and written communications skills through oral presentations (project presentation, department seminar or conferences, client interactions), technical reports and/or published research papers in conferences and/or journals (Communications);

- ❑ Use modern techniques, skills and tools necessary for computer science practices relevant to the project they undertake; use techniques in recent research papers to solve problems (Lifelong Learning).

HOW TO DO WELL IN THIS COURSE

- ❑ The students are advised to attend all their theory classes and respective labs regularly as both are integrated to each other. If any student will miss the theory lecture, he/she may not be able to do well in lab related to that topic.
- ❑ The students are advised to submit the assignments given in theory and lab classes timely to their respective Teachers/Instructors online.
- ❑ The students should demonstrate disciplined and well behaved demeanor in the Department.
- ❑ Each student shall be assigned a system in their introductory lab. They are advised to do their work on that system only for the whole semester. Students should store all their lab activities regularly.
- ❑ All students are advised to understand course objectives and outcomes and achieve both during their lab work.
- ❑ The students are advised to follow books/eBooks/online tutorial/other online study material links given in lecture/lab manual/ syllabus references. These study materials are very helpful in terms of skills, knowledge and placement.
- ❑ This lab course is very important in terms of placement. Therefore, students are advised to implement all the problems by her /him given in the individual week.
- ❑ All students are advised to solve old placement papers for campus selection. Following links may be useful for the preparation of your campus placements.

➤ <https://www.indiabix.com/placement-papers/companies/>

- <https://www.offcampusjobs4u.com/download-tcs-placement-test-question-papers-with-solutions/>
- <https://www.indiabix.com/placement-papers/tcs/>
- <https://www.firstnaukri.com/career-guidance/infosys-placement-papers-with-solutions-2019-firstnaukri-prep>
- <https://prepinsta.com/ibm/>
- <https://www.faceprep.in/infosys/infosys-aptitude-questions/>
- <https://alpingi.com/infosys-placement-papers-solution-pdf-download/>
- <http://placement.freshersworld.com/>

❑ The Students are advised to follow below tutorials' links:

- https://www.w3schools.com/php/php_intro.asp
- <https://www.w3schools.com/js/default.asp>
- <https://www.w3schools.com/sql/default.asp>
- <https://www.tutorialspoint.com/php/index.htm>
- <https://www.geeksforgeeks.org/php/>
- <https://www.geeksforgeeks.org/sql-tutorial/>
- <https://www.geeksforgeeks.org/javascript-tutorial/>

❑ The Students are advised to follow below Links for installing application software:

- <https://www.eclipse.org/pdt/>
- <http://php.net/manual/en/install.php>
- <https://docs.microsoft.com/en-us/dotnet/framework/install/guide-for-developers>
- <https://httpd.apache.org/download.cgi>
- <https://docs.microsoft.com/en-us/sql/database-engine/install-windows/install-sql-server?view=sql-server-ver15>
- <https://www.sqlservertutorial.net/install-sql-server/>

❑ The Students are advised to follow below Links for online compilers :

- https://www.onlinegdb.com/online_php_interpreter

- <http://phpfiddle.org/>
- https://www.tutorialspoint.com/execute_sql_online.php
- https://www.onlinegdb.com/online_sqlite_editor
- https://www.onlinegdb.com/online_csharp_compiler

❑ Skill Set that are required to develop by the students of MCA:

- Good communication and behavioral skills
- A positive attitude
- Confidence
- Strong technical skills
- Good command over programming languages like C, C++, Java, .Net, etc.
- Good programming skills and hands on experience
- Knowledge of data structure and database
- Awareness of latest technology trends.

TEACHING METHODS AND ASSESMENTS FOR ACHIEVING LEARNING OUTCOMES

This lab class will meet thrice per week for 100 minutes each meeting -- some lab class time may be traditional lectures, reviewing concepts and tools that are useful for the mini project, but most lab class time will be used for guided discussion and development, student presentations, and some team meetings. However, some lab classes may be used to discuss and solve subject related lab assignments given by the respective teachers of courses.

Generally, students will be taking up mini projects individually. However, in situations when they are working in teams, the individual responsibilities should be planned and documented throughout the phases of the project.

Students are expected to choose an appropriate project topic in consultation with the teacher, and do a short presentation that "pitches" the idea to the teacher and the class. While there is some flexibility in project selection, students should keep in mind the "capstone" nature of this class. Students must develop projects that

demonstrate that they have a working knowledge of basic and advanced concepts in computer science and also demonstrate a reasonable knowledge of recent developments in computer science. Each project should include non-trivial software development that has been approved by the teacher/instructor.

With an approved project, students will proceed through a standard sequence of software development stages, beginning with a requirements analysis and specification, and concluding with a final evaluation. A complete detail of the all project stages is given in the milestones section as well as summarized in "TOPICAL CALENDAR" section. At the end of each stage, each individual/team must produce a written report giving stage-specific documentation and describing the work performed, problems encountered, and decisions made. For team projects, the report must include a meeting log and breakdown of tasks by team member. One week before the completion of each stage, there will be a presentation from each project that previews the progress and results in that phase, for in-class discussion and suggestions for refinement in the following week. For these intermediate stage presentations, team members will rotate through as "presenter" for the team, and each student must make at least two intermediate-stage presentations (for a 3-person team this means that there will have to be multiple presentations on the same stage).

In the case of a group project, each member of the group must present the entire project, highlighting their individual contributions toward the project's success, and a short summary of each individual's contributions should be included in the final report as well.

SUBMISSION OF DELIVERABLES

Final project report, including all the deliverables, is required to be submitted strictly as per notified schedule.

EVALUATION AND GRADING

Students work on a single project throughout the duration of this course, and their course grade is calculated based on the grades for individual aspects and

milestones. The project will be graded for completeness, content, correctness, quality of presentation (oral and written reports), team work (in case of group project), and the demonstration of the student's knowledge in the computer science field.

As per the University norms Mini Project Report shall be finally evaluated by the external examiner at the end of the semester. However, there will be continuous monitoring of the progress and evaluation of the Mini Project during the semester and the distribution of marks shall be as follows:

Proposal	5 %
Presentations 1 & 2	
Presentations-1: Proposal/Synopsis	15 %
Presentations-II: SRS & Design	
Progress Report 1	
Requirements/Specification and Planning and Analysis	20 %
Progress Report 2	
System/Research Design	10 %
Progress Report 3	
Implementation and Testing	10 %
Final Deliverables:	
Final Presentation	10 %
Technical Report (including final source code)	30 %

REQUIRED TEXTS/READING/REFERENCES

Readings and references are project-specific, and will be determined by students/project groups, with approval of the teacher. All the resources used should be properly referenced.

Students will be making extensive use of external references for their project, and should be vigilant in maintaining high standards with regard to attribution and avoidance of plagiarism. If there are questions about how to deal with any such matters, the student should discuss the matter with the teacher concerned to make sure there are no misunderstandings.

ATTENDANCE POLICY

Attendance is vital for this class, since discussions, regular oral presentations and progress reports will have a strong impact on the ability to complete the project. You may be dropped from the course for missing more than two consecutive scheduled meetings/presentations.

LATE POLICY

Late work will not be accepted. In case of any unavoidable situations, make requests with the teacher/instructor to reschedule the assigned work/task on case to case basis, if possible.

MILESTONES FOR MINI-PROJECT

1. *Deciding and Registering the topic/title of the mini-project*

All the students are required to decide the topic/title for *real life software project*, which they want to design, develop and implement. In finalizing the proposed work topic, they may take help from concerned teachers/mentor in the lab. Decided topic/title *needs to be approved* by the concerned teacher/mentor in the lab.

Parallel Activity: Keep preparing the brief summary (synopsis/proposal) of the proposed work as per the given format.

2. *Submission of brief summary (synopsis/proposal) of the proposed work (As Per the Prescribed Format)*

Once the project topic/title is decided and approved, all students are required to **submit** and **present** a brief summary of the proposed work

(synopsis/proposal), clearly specifying the client's requirements for which the application software is being developed and the main features of the proposed software. After incorporating the suggestions of the teachers, if any, the final version of the summary of the proposed work (synopsis/proposal) should be submitted to concerned teacher in the lab.

3. *Submission Requirement specifications, planning and analysis*

After the submission and presentation of summary of the proposed work (synopsis/proposal), the students need to submit progress report-1, which includes: analysis modeling and related diagrams (please refer to the Topical calendar for more insights).

4. *System Analysis and Design Phase (Submission of SRS document)*

All students are required to study and analyze the present system (or existing manual system or proposed system), and all the findings should be **submitted** and **presented** in the form of *SRS document* along with *gantt chart* (using appropriate gantt tool, like GanttProject). While doing so, they may actively be involved with client, and/or teachers/Mentors for discussion. Some of the templates/formats of the typical *SRS document* are being attached for your reference.

Refer for SRS template:

- <http://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
- www.cse.msu.edu/~chengb/RE-491/Papers/SRSEExample-webapp.doc

Discuss the *SRS document* with the concerned teacher/Mentor in the lab, incorporate suggestions (if any), and maintain the different versions of SRS document. Students are required to present and submit *the final signed version of SRS document* to the concerned teacher/Mentor in the lab.

SRS document should contain the ER diagram, Data Flow Diagram and Data Dictionary. You should also prepare important UML diagrams like Use Case

diagrams, Activity diagrams, class diagrams, behaviour model and/or state transition diagram etc.

Students are advised to use standard tools for drawing UML diagrams, DFD, ER Diagrams, etc. Examples of some typical tools are listed below:

- UML diagrams using automated tool such as StarUML, BOUML etc.
- Data Flow Diagram (DFD) with different levels using tools such as Lucidchart, Visual Paradigm, etc.
- E-R Diagram with the help of automated tools such as ERDPlus, Smartdraw, etc.

Parallel Activity: In the mean time, you may learn and practice the tools necessary to develop the proposed software, and finalize the detailed database design, including populated tables. Also students should start the coding in parallel with their presentation of *SRS document*.

5. *System Development (Coding/Testing)*

Start the development of the system as per the design specifications discussed in *SRS document*. Coding should be well documented. **Technical Report** specifying the brief technical specifications and documenting the working of each major modules/methods should be submitted. Students are required to properly maintain the following during the system development:

- A clear design of working database of the system using a popular DBMS such as ORACLE, SQL Server, MYSQL, etc.

Deployment/Implementation

Deploy/Implement the developed system on the client site (actual user site) and obtain user acceptance letter, specifying that the developed system is working satisfactorily and is as per the specified requirements. Better you prepare an installation copy for your software along with installation manual.

Parallel Activity: Keep writing and preparing the **Final project Report** as per the standard format. (**Refer:** [Format for Project Report.pdf](#))

6. *Final Evaluation*

Demonstrate the working of system to the audience (teacher, Mentor students, clients), specifying the design and main features of the developed system. The **final project report** and an **oral presentation** should be submitted as per standard project report format/template. The user manual must be a part of the final project report and should be written in explanatory manner so that anyone can operate the system using this manual. Hardcopy as well as softcopy of the all the reports (like SRS Document, Technical report, Final Project report) should be submitted to the concerned teacher/Mentor in the lab. Softcopy of complete project (code), database and necessary files (preferably, installation software, along with installation manual) should also be submitted (*Refer: [Format for Project Report.pdf](#)*).

TOPICAL CALENDAR

S.No.	Project Stage/Activities	Deliverables	Duration	Latest By (Date)
01	Deciding and Registering the Topics/Titles of the Mini project	Registration of the Project topic/title Parallel Activity: Synopsis preparation	1-2 Week	
02	Brief Summary (Synopsis/Proposal) of the proposed work	Submission of Preliminary Proposal/Synopsis	2-3 Week	
		Presentation 1 (Proposal/Synopsis) Parallel Activity: Requirement Specifications and Planning and Analysis	3-4 Week	
03	Requirement Specification, Planning and Analysis	Progress Report 1 Submission <ul style="list-style-type: none"> • Approach and System Profile • Uses Cases • Feasibility and Draft Model • System and algorithm analysis • Preliminary object/process model • Tool selection, etc. 	5-7 Week	
04	System/Research Design	Progress Report 1 Submission (SRS & Design Document)	7-8 Week	
		Presentation 2 (SRS & Design) Parallel Activity: Implementation, Deployment and Testing	8-9 Week	
05	Implementation, Deployment and Testing	Progress Report 2 Submission (Technical Report) <ul style="list-style-type: none"> • Includes brief code walkthrough • Source code • Test results and discussion 	9-12 Week	
06	Evaluation and Refinement	Presentation 3 (Final presentation) Parallel Activity: Evaluation and Refinement (Final Report)	13-14 Week	
		Final Report submission	14 Week	