Savitribai Phule Pune University, Pune

Faculty of Commerce and Management

Bachelor of Commerce – Computer Application (BCom-CA)

Revised Curriculum (2024 Pattern as per NEP-2020)
w.e.f. Academic Year: 2026-2027

TYBCom-CA Semester V						
Course Type	Course	Course Code	Course Title	Credits		Hours / Week
				Theory	Practical	
Major Mandatory (10)	Major Mandatory 11	MJ-301-CA	Java Programming	4	-	4
	Major Mandatory 12	MJ-302-CA	Python Programming	4	-	4
	Major Mandatory 13	MJ-303-CA	Computer Laboratory based on Java and Python Programming	-	4	8
Major Elective	Major Elective 1	ME-301-CA	Essentials of Cloud Computing	4	-	4
Minor	Minor 4	MN-301-CA	Software Engineering	2	-	2
Vocational Skill Development Course (VSC)	Vocational Skill Development Course (VSC) (Practical)	VSC-301-CA	Entrepreneurship Essentials	-	2	4
Field Projects (FP)/ Community Engagement and Service corresponding to the Major (CEP)	Project	FP-301-CA	Project based on Major Mandatory	-	2	4
			Total	14	8	

Course Code	Type of Course	Title of the Course	Credits	Lectures Hours/Week
MJ-301-JP	Major Mandatory	**Java Programming**	4	4

Course Objectives:

- To introduce students to object-oriented programming concepts using Java.
- To develop programming skills for designing and implementing Java applications.
- To understand classes, objects, inheritance, polymorphism, and exception handling.
- To provide knowledge of Java GUI programming and event handling.
- To familiarize students with file handling and package creation in Java.
- Implement enterprise-level application development concepts.
- To build database-integrated applications using JDBC.
- To develop enterprise applications using modern frameworks like Spring.

Course Outcome: After completion of the course, students will be able to:

1. Design and develop OOP-based Java applications.
2. Apply collections and multithreading concepts.
3. Develop database-driven applications using JDBC.
4. Build enterprise applications using Servlet, JSP and Spring framework.

Unit	Title and Contents	No. of Lecture Hours
1	**An Introduction to Java** 1.1 Features of Java 1.2 Java Virtual Machine (JVM) 1.3 Java Development Kit (JDK) 1.4 Structure of a Java Program 1.5 Data Types 1.6 Variables 1.7 Operators 1.8 Control Statements 1.9 Arrays 1.10 Command Line Arguments.	**8**

2	**Object-Oriented Programming Concepts** 2.1 Classes and Objects 2.2 Constructors and method overloading 2.3 Recursion and returning objects 2.4 Static, this keyword 2.5 Nested and inner classes 2.6 String handling and wrapper classes 2.7 Inheritance and method overriding 2.8 Abstract Classes and Interfaces 2.9 Access control	**8**
3	**Packages & Collection** **Packages:** 3.1 Packages concept 3.2 Creating user-defined packages 3.3 Java Built-in Packages 3.4 Import Statement, Static Import **Collection:** 3.5 Introduction to the Collection framework 3.6 List – ArrayList 3.7 LinkedList 3.8 Set - HashSet, TreeSet 3.9 Map - HashMap and TreeMap 3.10 Interfaces such as Comparator, Iterator, ListIterator, Enumeration	**10**
4.	**File & Exception handling** **Exception**: 4.1 Exception and Error 4.2 Use of try, catch, throw, throws, finally 4.3 Built and Custom Exceptions 4.4 Throwable Class **File handling:** 4.5 Overview of different streams (Byte streams and character streams) 4.6 Readers, Writers Class 4.7 File Class 4.8 File Input Stream, File Output Stream 4.9 Input Stream Reader and Output Writer class 4.10 FileReader and FileWriter classes 4.11 Buffered Reader class	**10**

5	**Multithreading & Networking** **Multithreading:** **5.1** Introduction to Multithreading **5.2** Thread creation: Thread Class, Runnable Interface. 5.3 Life cycle of Thread. 5.4 Thread Priority. 5.5 Execution of Thread Application. 5.6 Synchronisation and Inter-thread Communication. **Networking:** 5.7 Overview of Networking. 5.8 Networking Basics: Port Number, Protocols and Classes. 5.9 Sockets, reading from and writing to a Socket.	**8**
6	**User Interface with AWT and Swing** 6.1 What is AWT? 6.2 What is Swing? 6.3 Difference between AWT and Swing 6.4 The MVC Architecture and Swing 6.5 Layouts and Layout Managers 6.6 Containers and Components – JFrame, JButton, JLabel, JText, JTextArea,JCheckBoxandJRadioButton, JList, JComboBox, JMenuand related Classes 6.7Dialogs (Message, Confirmation, Input), JFileChooser, JColorChooser 6.8 Event Handling: Event Sources, Listeners 6.9 Adapters, and Anonymous Inner Class	**10**
7	**JDBC** 7.1 Introduction 7.2 JDBC Architecture. 7.3 JDBC Process 7.4 Working with ResultSet Interface	**6**

Internship for Students if any:

List of Recommended Books and Study Materials

Reference Books

Sr. No.	Title of the Book	Author/s	Publication	Place
1	Head First Java	Kathy Sierra & Bert Bates	O'Reilly Media	Sebastopol, California, USA
2	Java: The Complete Reference	Herbert Schildt	McGraw-Hill Education	New York, USA
3	Core Java Volume I – Fundamentals	Cay S. Horstmann & Gary Cornell	Prentice Hall	Upper Saddle River, New Jersey, USA
4	Java Concurrency in Practice	Brian Goetz et al.	Addison-Wesley	Boston, Massachusetts, USA

5	Spring in Action	Craig Walls	Manning Publications	Shelter Island, New York, USA

E- Resource:

Oracle Java Tutorials –
https://docs.oracle.com/javase/tutorial/
Spring Framework Documentation –
https://spring.io/projects/spring-framework
Hibernate Documentation –
https://hibernate.org/orm/documentation/
GeeksforGeeks Java Tutorials –
https://www.geeksforgeeks.org/java/

Course Code	Type of Course	Title of the Course	Credits	Lectures Hours/Week
MJ-302-PP	Major Mandatory	**Python Programming**	4	4

Course Objectives

1. Understand the fundamentals of Python programming
2. Apply Python data structures, functions, modules, and packages to build modular applications.
3. Perform file operations and implement exception handling mechanisms for robust programs.
4. Utilize Python libraries for data analysis, visualization, and business data processing.
5. Develop basic web applications using the Flask framework and understand web application architecture.

Course Outcomes: After completion of the course, students will be able to:

CO1: Demonstrate proficiency in Python syntax, control structures, functions, and built-in data structures.
CO2: Design and implement modular Python programs using modules and packages.
CO3: Develop applications that perform file handling operations and manage runtime errors through exception handling.
CO4: Analyze and manipulate datasets using Python libraries such as NumPy, Pandas, and Matplotlib.
CO5: Create basic web applications using Flask and implement routing, templates, and form handling.
CO6: Apply Python programming skills to develop solutions for business, analytics, and web-based applications.

Unit	Title and Contents	No. of Lectures
1	**Fundamentals of Python** 1.1 Introduction to Python 1.1.1 History and Evolution of Python 1.1.2 Features and Applications of Python 1.1.3 Installation of Python and Setting up the Path 1.1.4 Working with Python Interpreter and IDLE 1.1.5 Basic Syntax and Program Structure 1.1.6 Variables and Data Types 1.1.7 Operators and Expressions 1.2 Decision Making and Iterative Statements 1.2.1 Conditional Statements: if, if-else, Nested if-else, if-elif-else 1.2.2 Looping Statements: for, while, Nested Loops 1.2.3 The range() Function 1.2.4 Control Statements: break, continue, pass 1.3 String Manipulation 1.3.1 Introduction to Strings 1.3.2 Accessing String Elements 1.3.3 String Indexing and Slicing 1.3.4 Basic String Operations 1.3.5 String Functions and Methods 1.4 Lists 1.4.1 Introduction to Lists 1.4.2 Creating and Accessing Lists 1.4.3 List Operations 1.4.4 Working with Lists	**12**

	1.4.5 List Functions and Methods 1.5 Tuples 1.5.1 Introduction to Tuples 1.5.2 Creating and Accessing Tuples 1.5.3 Tuple Operations 1.5.4 Tuple Functions and Methods 1.6 Dictionaries 1.6.1 Introduction to Dictionaries 1.6.2 Creating and Accessing Dictionary Elements 1.6.3 Working with Dictionaries 1.6.4 Properties of Dictionaries 1.6.5 Dictionary Functions and Methods 1.7 Sets 1.7.1 Introduction to Sets 1.7.2 Creating and Accessing Set Elements 1.7.3 Set Operations 1.7.4 Working with Sets - Properties,methods 1.7.5 Properties of Sets 1.8 Functions 1.8.1 Defining and Calling Functions 1.8.2 Types of Functions 1.8.3 Function Arguments 1.8.4 Anonymous (Lambda) Functions 1.8.5 Global and Local Variables	
2	**Modules and Packages** 2.1 Introduction to Modules 2.1.1 Concept of Modules · Need and advantages of modules · Types of modules: 2.1.2 Importing Modules in Python 2.1.3 Working with Built-in Modules · Random Module · Math Module · Time Module · Datetime Module · Calendar Module · Sys Module 2.2 User Defined Modules and Functions 2.2.1 User Defined Functions · Defining functions · Function parameters and return values · Scope of variables · Recursive functions 2.2.2 Structure of Python Modules · Creating a module · Saving module files · Importing user-defined modules · Using module variables and functions · __name__ == "__main__" concept 2.2.3 Module Documentation · Docstrings · Comments	**8**

	· Using help() on custom modules 2.3 Packages in Python 2.3.1 Introduction to Packages · Need for packages · Package hierarchy · __init__.py · Difference between module and package 2.3.2 User Defined Packages · Creating packages · Creating sub-packages · Importing modules from packages · Relative and absolute imports	
3	**File Handling** 3.1 Reading and writing Files 3.1.1 Introduction to File Handling 3.1.2 Opening a File 3.1.3 Reading Data from Files 3.1.4 Writing Data to Files 3.1.5 Closing Files 3.2 Text and Binary Files 3.2.1 Text Files 3.2.2 Reading JSON Files 3.2.3 Writing JSON Files 3.2.4 CSV vs JSON 3.2.5 Binary Files 3.2.6 Difference Between Text and Binary Files	**6**
4.	**Exception Handling** 4.1 Python Exception 4.1.1 Introduction to Exceptions 4.1.2 Exception Hierarchy 4.1.3 Causes of Exceptions 4.1.4 Importance of Exception Handling 4.2 Common Exception 4.2.1 Arithmetic Exceptions 4.2.2 Type and Value Exceptions 4.2.3 Index and Key Exceptions 4.2.4 Name and Attribute Exceptions 4.2.5 File-Related Exceptions 4.2.6 Import Exceptions 4.3 Exception handling in Python (try-except-else) 4.3.1 The except statement with no exception 4.3.2 Multiple Exception 4.3.3 The try-finally clause 4.3.4 Custom Exception and assert statement	**4**
5	**5.1 Introduction to Data Analytics and Data Visualization** 5.1.1 Overview of Data Analytics- Data Analytics Lifecycle, Applications of Data Analytics in Business, Role of Python in Data Analytics 5.1.2 Data Cleaning and Transformation- Importance of Data Cleaning, Data Cleaning Techniques, Removing Duplicates, ata Transformation Concepts 5.1.3 Introduction to Data Visualization - Importance of Data	**15**

	Visualization, Types of Data Visualizations, Choosing Appropriate Charts 5.1.4 Popular Python Libraries for Data Analytics - NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn. Plotly **5.2 NumPy for Numerical Computing and Data Processing** Introduction to NumPy - Need and Features of NumPy, Advantages of NumPy over Python Lists, Importing NumPy, NumPy Arrays **5.3 Pandas for Data Manipulation and Analysis** Introduction to Pandas - Need and Features of Pandas, Series, DataFrameFunctions - pd.Series(), pd.DataFrame(), **5.4 Data Visualization using Matplotlib** Fundamentals of Matplotlib - Introduction to Matplotlib, Importing Matplotlib, Figure and Axes Concepts, Functions - plt.figure(), plt.show(), Line Chart, Bar Chart, Pie Chart	
6	**6.1 Introduction to Flask** 6.1.1 Introduction to Web Development 6.1.2 Client-Server Architecture 6.1.3 Introduction to Flask Framework 6.1.4 Installation of Python and Flask 6.1.5 Creating First Flask Application **6.2 Routing and Templates** 6.2.1 URL Routing 6.2.2 Dynamic Routes 6.2.3 HTML Templates using Jinja2 6.2.4 Passing Data to Templates **6.3 Forms and User Input** 6.3.1 HTML Forms 6.3.2 GET and POST Methods 6.3.3 Handling Form Data in Flask 6.3.4 Form Validation Basics **6.4 Database Connectivity** 6.4.1 Introduction to SQLite 6.4.2 Connecting Flask with SQLite 6.4.3 CRUD Operations (Create, Read, Update, Delete) 6.4.4 Displaying Database Records	**15**

List of Recommended Books and Study Materials

1. Python Crash Course – Eric Matthes, No Starch Press.
2. Learning Python – O'Reilly Media.
3. Python Programming: Using Problem Solving Approach – Oxford University Press.
4. Core Python Programming – Dreamtech Press.
5. Python for Data Analysis – O'Reilly Media.
6. Automate the Boring Stuff with Python
7. Flask Web Development
8. Introduction to Machine Learning with Python
9. Data Science from Scratch
10. Python for Data Analysis – Wes McKinney
11. Python Data Science Handbook – Jake VanderPlas
12. Data Analysis Numpy, Matplotlib and Pandas by Bernd Klein
13. Python Data Analytics Data Analysis and Science Using Pandas, matplotlib, and the Python Programming Language Fabio Nelli

Suggested Online Resources / Video Lectures

Data Analytics using Python

1. **Kaggle Learn Courses**
   - **Free micro-courses on Python, Pandas, and Data Visualization.**
2. **DataCamp Free Resources**
   - **Tutorials on Pandas, NumPy, and analytics concepts.**
3. **Analytics Vidhya Learning Hub**
   - **Indian platform with practical examples and datasets.**

Flask Web Development

1. **Real Python Flask Tutorials**
   - **Excellent step-by-step Flask tutorials.**
2. **Python Engineer Flask Tutorials**
   - **Beginner-oriented Flask projects.**
3. **DigitalOcean Flask Tutorials**
   - **Practical examples with deployment concepts.**

Course Code	Type of Course	Title of the Course	Credits	Lecture Hours/Week
ME-301-ECC	**Major Elective**	**Essentials of Cloud Computing**	**4**	**4**

Course Objectives:

1. To introduce students to the fu ndamental concepts, characteristics, and service models of cloud computing.
2. To enable students to understand and differentiate various cloud deployment models (Public, Private, Hybrid, Community).
3. To familiarize students with major cloud service providers and their core offerings.
4. To develop understanding of cloud storage, virtualization, security, and resource management.
5. To equip students with practical knowledge of cloud-based tools and applications used in industry.

Course Outcomes: After completion of the course, students will be able to:

CO1: Explain the fundamental concepts, characteristics, and evolution of cloud computing.

CO2: Compare and evaluate different cloud service models (IaaS, PaaS, SaaS) and deployment models.

CO3: Understand the role of virtualization in cloud computing and differentiate between VMs and containers.

CO4: Analyze cloud storage architectures and apply data management strategies in a cloud environment.

CO5: Assess cloud security risks and apply appropriate security mechanisms and compliance standards.

CO6: Identify emerging trends in cloud computing and evaluate career pathways in the cloud domain.

Unit	Title and Contents	No. of Lectures
I	**Introduction to Cloud Computing** Definition, Characteristics, and Evolution of Cloud Computing; Cloud vs. Traditional IT Infrastructure; Benefits and Challenges of Cloud Adoption; Cloud Computing Architecture (Front-end, Back-end, Network); Overview of Cloud Service Models: IaaS, PaaS, SaaS; Real-world Cloud Applications 1. Cloud storage (Google Drive, OneDrive) 2. Online collaboration tools (Google Workspace, Microsoft 365) 3. Cloud applications in education, business, healthcare, and banking	15
II	**Cloud Deployment Models & Virtualization** Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud; Advantages and Disadvantages of Each Model; Introduction to Virtualization: Concept and Types (Full, Para, OS-Level); Hypervisors: Type 1 and Type 2; Virtual Machines vs. Containers; Introduction to Docker and Kubernetes (Overview); Resource Pooling and Multi-Tenancy.	15
III	**Cloud Storage and Data Management** Cloud Storage Architecture and Types (Block, Object, File Storage); CAP Theorem; Data Replication and Redundancy; Cloud Databases: SQL vs. NoSQL in Cloud; Big Data and Cloud Integration; Data Migration to Cloud; Cloud CDN (Content Delivery Network); Overview of AWS S3, Google Cloud Storage, Azure Blob Storage.	15
IV	**Cloud Security, Compliance & Emerging Trends** Cloud Security Fundamentals: Threats, Risks, and Best Practices; Identity and Access Management (IAM); Data Encryption in Cloud; Compliance and Legal Issues (GDPR, IT Act); SLA (Service Level Agreements); Emerging Trends: Serverless Computing (FaaS), Edge Computing, Fog Computing, AI/ML as a Cloud Service; Introduction to Multi-Cloud Strategy; Career Opportunities in Cloud Computing.	15

Internship for Students if any:

Students may undertake a short-term internship / virtual internship with cloud providers (AWS Educate, Google Cloud Skills Boost, Microsoft Azure Fundamentals) or cloud-based IT firms. Minimum duration: 4 weeks.

List of Recommended Books and Study Materials

1. Rajkumar Buyya, James Broberg, Andrzej Goscinski – Cloud Computing: Principles and Paradigms, Wiley, 2011.

2. Michael Miller – Cloud Computing: Web-Based Applications That Change the Way You Work, Que Publishing.

3. Thomas Erl, Ricardo Puttini, Zaigham Mahmood – Cloud Computing: Concepts, Technology & Architecture, Prentice Hall.

4. NPTEL/Swayam Online Course: Cloud Computing by Prof. Soumya Kanti Ghosh, IIT Kharagpur.

5. Official Documentation: AWS Cloud Practitioner, Google Cloud Fundamentals, Microsoft Azure Fundamentals (AZ-900).

Course Code	Type of Course	Title of the Course	Credits	Lectures Hours/Week
MN-301-SE	**Minor**	**Software Engineering**	**2**	**2**

Course Objectives	
1	To provide knowledge of fundamental software engineering principles, methodologies, and best practices used in software development and to introduce students to the concepts of systems, software, and the role of software engineering in developing quality software solutions
2	To develop an understanding of Software Development Life Cycle (SDLC) models and their application in software project management and to equip students with the skills required for requirement elicitation, analysis, specification, and management in software projects.

Course Outcome **Student will be able to**	
CO1	Apply fundamental software engineering principles and system concepts to understand software characteristics and the role of software engineering in modern organizations.
CO2	Analyze and compare SDLC models and apply requirement engineering techniques to identify, document, and manage software requirements.

Unit	Title and Contents	No. of Lectures

1	**Fundamentals of Software Engineering and System Concept** 1.1 Basics of Software and Software Engineering 1.1.2 Introduction to software Engineering 1.1.3Definition of Software 1.1.3 The evolving Role of Software 1.1.4 Software Characteristics 1.1.5 Software Components 1.1.6 Software Applications 1.1.7 Software Application Domain 1.1.8 Types of Software 1.1.9 Software Crisis-Problem and Causes 1.2 System Concept 1.2.1 Definition 1.2.2 Basic Components 1.2.3 Elements of the system 1.2.4 Types of System 1.2.5 System Characteristics	15
2	**Software Development Life Cycle (SDLC) & Software Requirement Analysis** 2.1 Introduction 2.2 Activities of SDLC 2.3 Software Development Life Cycle(SDLC) 2.4 Importance of SDLC 2.5 Embedding Security into the SDLC 2.6 Real Life Example of SDLC 2.7 SDLC Models 2.7.1 Prescriptive Process Model 2.7.2 Waterfall Model 2.7.3 Incremental Process Models 2.7.4 Evaluationary Process Model 2.7.5 Prototyping Model 2.7.6 Spiral Model 2.7.7 Concurrent Model 2.8 Requirement Engineering 2.8.1 Introduction to Requirement Engineering 2.8.2Requirement Engineering Process 2.8.3Requirement Elicitation Techniques	15

Course Outcome:

CO1: Apply fundamental software engineering principles, techniques, and practices to software development activities and understand the system concepts, software characteristics, and the significance of software engineering in modern organizations

CO2: Analyze and compare different SDLC models and select appropriate models for various software

development scenarios and identify, gather, document, and manage software requirements using requirement engineering techniques and tools.

List of Recommended Books and Study Materials
1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, Bruce R. Maxim, McGraw-Hill Education, New York, USA 2. Software Engineering, Ian Sommerville, Pearson Education, London, UK 3. Software Engineering: A Practitioner's Approach, Roger S. Pressman, Bruce R. Maxim, McGraw-Hill Education, New York, USA 4. Software Engineering Principles and Practice, Hans van Vliet, Wiley India, New Delhi. 5. Object-Oriented and Classical Software Engineering, Stephen R. Schach, McGraw-Hill Education, New York, USA 6. Software Engineering Concepts, Richard Fairley, McGraw-Hill Education, New York, USA 7. Software Engineering: Principles and Practice, Waman S. Jawadekar, McGraw-Hill Education, New Delhi. **E-Learning Resource Website Links for Software Engineering** 1. **NPTEL (National Programme on Technology Enhanced Learning)** NPTEL 2. **SWAYAM Online Courses** SWAYAM 3. **Geeks for Geeks – Software Engineering** Geeks for Geeks Software Engineering 4. **IBM Skills Build** IBM Skills Build

Course Code	Type of Course	Course Title	Credits	Hours/ Week
VSC-301-EE	Vocational Skill Development Course (VSC)	Entrepreneurship Essentials	2	4

Course Objectives	
1	To understand the fundamentals of entrepreneurship and entrepreneurial leadership.
2	To develop the ability to identify customer needs and business opportunities.

Course Outcome	
CO1	To explain the concepts and principles of entrepreneurship and entrepreneurial leadership.
CO2	To analyse customer requirements and identify potential business opportunities.

Unit	Title and Contents	No. of Lecture Hours
1 Entrepreneurship Fundamentals	1.1 Meaning and concept, attributes and mind-set of entrepreneurial and intrapreneurial leadership, role models in each and their role in economic development. 1.2 Understanding and analyzing the macro-Problem and Industry perspective, technological, socio economic and urbanization trends and their implication on new opportunities. 1.3 Aligning passion, identifying and defining problem using design thinking principles, analyzing problem and validating with the potential customer.	15
2 Customer Discovery, Ideation and competition mapping	2.1 Understanding customer, customer segmentation, creating and validating customer personas. 2.2 Understanding Customer Jobs-to-be-done and crafting innovative solution design to map to customer's needs and create a strong value proposition. Iterating problem-customer fit. 2.3 Examining ideation techniques and generating solution ideas. Competition and Industry trends mapping for assessing market sizing - initial opportunity.	15

This Course will be executed in collaboration with SPPU's Innovation, Incubation & Linkages and Wadhavani Foundation.

Reference Material:

Reference Books

Sr. No.	Title of the Book	Author/s	Publication	Place

1	Entrepreneurship (11th Edition)	Robert D. Hisrich, Michael P. Peters, Dean A. Shepherd, Sabyasachi Sinha	McGraw Hill	New York
2	The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses	Eric Ries	Crown Business	New York
3	Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers	Alexander Osterwalder & Yves Pigneur	John Wiley & Sons	Hoboken, New Jersey
4	Start with Why	Simon Sinek	Penguin Books Limited	London
5	Change by Design Revised & Updated: How Design Thinking Transforms Organizations and Inspires Innovation	Tim Brown	Harper Business	New York
6	The Dolphin and the Shark: Stories on Entrepreneurship	Namita Thapar	Penguin Books Limited	New Delhi
7	Effectuation: Elements of Entrepreneurial Expertise	Saras D. Sarasvathy	Elgar Publishing Ltd	Cheltenham, UK

Other Learning Material E- Resource:

1. https://onlinecourses.nptel.ac.in/noc25_mg53/preview
2. https://www.coursera.org/learn/wharton-entrepreneurship
3. https://open.umn.edu/opentextbooks/textbooks/entrepreneurship-and-innovation-toolkit
4. https://learn.saylor.org/course/view.php?id=72
5. https://www.tutorialspoint.com/entrepreneurship_development/index.htm

Course Code	Type of Course	Course Title	Credits	Hours/ Week
FP-301-MM	**Field Project**	**Project based on Major Mandatory**	**2**	**4**

Course Objectives

1. To enable students to apply programming and software development concepts from the Major Mandatory courses (Java / Python) to build a complete, functional application.
2. To develop the ability to identify a real-world problem, analyze requirements, and design an appropriate software solution.
3. To impart hands-on experience of the Software Development Life Cycle — design, implementation, and testing.
4. To develop teamwork, documentation, and technical communication skills through project work, reporting, and presentation.

Course Outcomes

After completion of the course, students will be able to:

CO1: Identify a real-world problem, conduct a literature survey, and formulate project objectives and scope.
CO2: Design and implement a software solution using appropriate system models (ERD, DFD) and programming skills.
CO3: Test and validate the developed system for functional correctness and reliability.
CO4: Prepare a project report conforming to academic standards and demonstrate the work through presentation and viva.

Guidelines:

- Students should work in a team of maximum 2 students.
- Students can choose a project topic without any restriction on technology or domain.
- The student group will work independently throughout the project work including: problem identification, information searching, literature study, design and analysis, implementation, testing, and the final reporting.
- Project guide must conduct project presentations (minimum 4) to monitor the progress of the project groups.
- At the end of the project, the group should prepare a report which should conform to international academic standards. The report should follow the style in academic journals and books, with clear elements such as: abstract, background, aim, design and implementation, testing, conclusion and full references, Tables and figures should be numbered and referenced to in the report.
- The final project presentation with demonstration (UE) will be evaluated by the project guide (appointed by the college) and one external examiner (appointed by the University).

Evaluation guidelines:

- Evaluation will be done on the following criteria
    - Internal Evaluation

- First Presentation
- Second Presentation
- Documentation

- External Evaluation
  - Project Presentation
  - Demonstration
  - Viva

Documentation Format:

1. **Abstract**
2. **Introduction**
   - 2.1 motivation
   - 2.2 problem statement
   - 2.3 purpose/objective and goals
   - 2.4 literature survey
   - 2.5 project scope
3. **System analysis**
   - 3.1 Existing systems
   - 3.2 Scope and limitations of existing systems
   - 3.3 project perspective, features
   - 3.4 stakeholders
   - 3.5 Requirement analysis -Functional requirements, performance requirements, security requirements etc.
4. **System Design**
   - 4.1 Design constraints
   - 4.2 System Model: ERD & DFD (Context Level)
   - 4.3 Data Model
   - 4.4 User Interface
5. **Implementation details**
   - 5.1 Software / hardware specifications
6. **System Outputs and Validation**
   - 6.1 Output Screen
   - 6.2 Input Validation
   - 6.3 Report Format
7. **Conclusion and Recommendations Limitations & Future Scope**
8. **Bibliography and References**