

Cummins College of Engineering for Women
(An autonomous institute affiliated to Savitribai Phule pune university)
Karve Nagar, Pune - 411 052.



Vision

To be globally renowned engineering institute for imparting holistic education and developing professional women leaders in engineering and technology

Syllabus

of

**T. Y. BTech
(Computer Engineering)**

2023 Pattern [R0]

List of Abbreviations

Abbreviation	Title
PCC	Programme Core Course
BSC	Basic Science Course
ESC	Engineering Science Course
PE	Programme Elective Course
OE	Open Elective
VSEC	Vocational and Skill Enhancement Course
CC	Co-curricular Courses / Liberal Learning Course
IKS	Indian Knowledge System
VEC	Value Education Course
RM	Research Methodology
INTR	Internship
PROJ	Project
CEP	Community Engagement Project
RM	Research Methodology
Mm	Multidisciplinary Minor
AEC	Ability Enhancement Course

Curriculum for UG Degree Course in BTech. Computer Engineering
(Academic Year: 2025-26 Onwards)

Third Year | Semester-VI

Course Code	Course Title	Teaching Scheme Hours / Week			Cr	Examination Scheme			Total Marks
		L	T	P		ISE	ESE	Pr/Or	
23PCCE601	Full Stack Development	3	0	0	3	50	50	00	100
23PCCE602	Cloud Computing	3	0	0	3	50	50	00	100
23PCCE603	Information and Cyber Security	3	0	0	3	50	50	00	100
23PCCE604	Data Engineering	3	0	0	3	50	50	00	100
23PECE601	Programme Elective-II	3	0	0	3	50	50	00	100
23MmCE601	Enterprise Banking and Insurance	2	0	0	2	25	25	00	50
23VSEC601L	Programming Skills Development Laboratory II	0	0	4	2	25	00	25	50
23PCCE601L	Full Stack Development Laboratory	0	0	2	1	25	00	25	50
23PCCE602L	Cloud Computing Laboratory	0	0	2	1	25	00	25	50
23PCCE604L	Data Engineering Laboratory	0	0	2	1	25	00	25	50
Total		17	00	10	22	375	275	100	750

L=Lecture, T=Tutorial, P= Practical, Cr= Credits, ISE =In Semester Evaluation, ESE =End Semester Examination, Pr/Or = Practical/Oral

23PECE601 Programme Elective-II

- A. DevOps Fundamentals
- B. Deep Learning
- C. Software Testing and Quality Assurance
- D. Distributed Systems


APPROVED BY
 Secretary Academic Council
 MKSSS's Cummins College of Engineering
 For Women, Pune-411052




APPROVED BY
 Chairman Academic Council
 MKSSS's Cummins College of Engineering
 For Women, Pune-411052

23PCCE601 FULL STACK DEVELOPMENT

Teaching Scheme

Lectures: 3 Hours / week

Examination Scheme

In Semester: 50 Marks

End Semester: 50 Marks

Credits: 3

Prerequisites: Computer Networks

Course Objectives

To facilitate the learner to

1. Get exposure to full stack development technologies
2. Develop familiarity with the client side technologies for Web development
3. Gain comprehensive knowledge about server side technologies for enterprise application development in practice
4. Get familiar with the web services based approach for real-life application development.
5. Get acquainted with the database development technologies

Course Outcomes

By taking this course, the learner will be able to

1. Choose suitable client side technologies for web application development
2. Apply server side technologies for enterprise application development
3. Analyze the characteristics of the web services paradigm
4. Make use of database development technologies to realize their suitability for application development

Unit I: Client Side Web Technologies

n-tier architecture, Hyper Text Transfer Protocol (HTTP) request - response, Web browser, Full stack development: overview and types, Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), Extensible Markup Language (XML), JavaScript (JS), Web formats: CSV, XML, JavaScript Object Notation (JSON), Document Object Model (DOM), Asynchronous JavaScript And XML (AJAX).

Unit II: Server Side Web Technologies

Role of server side technology, Common Gate Interface (CGI), Java Servlets, Java Server Pages (JSP), Session tracking, Java Beans, Model-View-Controller (MVC) architecture.

Unit III: Technologies for Enterprise Application Development

Overview of Java Enterprise Edition (JEE) technologies, Enterprise Java Beans, Java Messaging Service (JMS), Remote Method Invocation (RMI), Spring Framework-Fundamentals, Architecture, Application.

Unit IV: Web Services

Basics of Web Services, Service Oriented Architecture (SOA), Java Web services based on SOAP and Representational State Transfer (REST), Java Web services API for SOAP and REST based web services.

Unit V: Modern Frameworks

ReactJS Fundamentals, Features, Advantages, ReactJS concepts like components, virtual DOM, JavaScript XML (JSX) and APIs, Comparison with other frameworks like AngularJS, Introduction to Nodejs.

Unit VI: Database Programming and Object-Relational Mapping (ORM)

Database Connectivity (DBC), Transactions and consistency API - Java Transaction API (JTA), ORM: Need, architecture, advantages, Java Persistence API (JPA), ORM Framework - Hibernate - Features, architecture.
Case study such as Online Banking, E- commerce application.

Text Books

1. “Web Technologies:HTML, JS, PHP, JAVA, JSP, ASP.net, XML, AJAX Black Book”, Kogent Learning Solutions Inc., DreamTech Press, (2015) , ISBN: 978-81-7722-997-4.
2. “Java Server Programming Java EE7 Black Book”, Kogent Learning Solutions Inc., DreamTech Press, (2016), ISBN: 978-81-7722-936-3.
3. “The Complete Reference J2EE”,Jim Keogh, McGraw Hill Education,(2012) ISBN: 978-0-07-052912-0.
4. Stoyan Stefanov, “React - Up & Running: Building Web Applications”, O’Reilly,2nd Edition, (2021), ISBN: 9781491931820.
5. Philip Ackermann, “Full Stack Web Development Comprehensive Guide”,SPD,2nd Edition ,(2023),ISBN 978-1-4932-2437-1.

Reference Books

1. Mark Tielens Thomas, “React in Action”, Manning Publications,(2018), ISBN: 978-1617293856.
2. Kevin Mukhar, Chris Zelenak, James L. Weaver and Jim Crume, Apress,“Beginning Java EE5: From Novice to Professional”, (2006) ,ISBN 9798181284029

3. William Crawford, Jim Farley, "Java Enterprise in a Nutshell", O'Reilly, 3rd Edition, (2007) ISBN-13: 978-8-8404-287-8.
4. Jan Machacek, Aleksa Vukotic, J. Ditt, A. Chakraborty, Apress, "Pro Spring 2.5", 2nd Edition, (2010), ISBN 978-81-8489-054-9.

Online/Web/Other References

1. <https://docs.oracle.com/javaee/7/tutorial/> (Last access on 9 December 2025)
2. <https://reactjs.org> (Last access on 9 December 2025)
3. <https://www.springboottutorial.com/introduction-to-spring-boot> (Last access on 9 December 2025)
4. <https://www.springboottutorial.com/spring-boot-projects-with-code-examples> (Last access on 9 December 2025)

23PCCE602 CLOUD COMPUTING

Teaching Scheme

Lecture: 3 Hours / week

Examination Scheme

In Semester: 50 Marks

End Semester: 50 Marks

Credits: 3

Prerequisites: Operating Systems

Course Objectives

To facilitate the learner to

1. Understand the basic concepts related to cloud computing
2. Analyze the underlying principles of different cloud service models
3. Understand and apply the security techniques in cloud computing
4. Get exposure to emerging trends in cloud computing

Course Outcomes

By taking this course, the learner will be able to

1. Utilize cloud computing concepts and the emerging trends for cloud-based systems
2. Examine the cloud services and models for implementing cloud-based systems
3. Choose various cloud platforms and tools for realization of different services
4. Make use of security concepts in the cloud environment

Unit I: Basics of Cloud Computing

Introduction to Cloud Computing, Cloud Economics, National Institute of Standards and Technology (NIST) Definition of Cloud Computing, Cloud Characteristics, Service Level Agreement (SLA), Cloud Service Models, Cloud Deployment Models, Observability in Cloud Deployment, Benefits, Challenges and Risks.

Unit II: Infrastructure-as-a-Service (IaaS)

Preliminaries of Infrastructure-as-a-Service (IaaS), Virtualization – Introduction, Types, Characteristics, Pros and Cons, Hypervisors - Kernel Virtual Machine (KVM), Oracle VirtualBox, Docker Containers, Microservices. Case Study- Amazon Web Services (AWS).

Unit III: Platform-as-a-Service (PaaS)

Fundamentals of Platform-as-a-Service (PaaS), Data in Cloud: Relational Databases, NoSQL Databases, Big Data, Cloud File System: Hadoop Distributed File System (HDFS), HBase, Map-Reduce Model. Case Study - Google App Engine (GAE).

Unit IV: Software-as-a-Service (SaaS)

Overview of Software-as-a-Service (SaaS), Multi-tenancy, Mashups, Service Oriented Architecture (SOA), Web Services based on Simple Object Access Protocol (SOAP) and

REpresentational State Transfer (REST), SaaS Applications, Case Study - Salesforce.

Unit V: Cloud Security

Cloud Security Fundamentals, Cloud Security Challenges and Risks, Virtualization Security, Identity Management and Access Control, Secure Execution Environment and Communication, Data Privacy by Design (DPbD).

Unit VI: Recent Trends

Inter-cloud / Federated Cloud, Multi-cloud, Internet of Things (IoT) and Cloud Computing, Mobile and Cloud Computing, Data Centers, Cloud and DevOps, Serverless computing, Artificial Intelligence (AI) in Cloud Computing.

Text Books

1. Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, “Mastering Cloud Computing”, *McGraw Hill Education*, (2019), ISBN 13: 978-1-25-902995-0
2. Gautam Shroff, “Enterprise Cloud Computing”, *Cambridge University Press*, (2016), ISBN 13: 978-1-107-64889-0
3. Ronald Krutz and Russell Dean Vines, “Cloud Security”, *Wiley India Pvt. Ltd.*, (2018), ISBN 13: 978-81-265-2809-7
4. Kailash Jayaswal, Jagannath Kallakurchi, Donald Houde, Dr. Deven Shah, “Cloud Computing Black Book”, *DreamTech Press*, (2018), ISBN 13: 978-93-5119-418-7

Reference Books

1. Thomas Erl, Zaigham Mahmood and Ricardo Puttini, “Cloud Computing Concepts, Technology and Architecture”, *Prentice Hall*, (2018), ISBN 13: 978-93-325-3592-3
2. Barrie Sosinsky, “Cloud Computing Bible”, *Wiley India Pvt. Ltd.*, (2018), ISBN 978-81-265-2980-3
3. Rajkumar Buyya, James Broberg, Andrzej Goscinski, “Cloud Computing Principles and Paradigms”, *Wiley India Pvt. Ltd.*, (2017), ISBN 13: 978-81-265-4125-6
4. Dr. Kumar Saurabh, “Cloud Computing”, *Wiley India Pvt. Ltd.*, 4th Edition, (2018), ISBN 978-81-265-7096-6
5. Tim Mather, Subra Kumaraswamy, Shahed Latif, “Cloud Security and Privacy”, *O'Reilly*, (2018), ISBN 13: 978-81-8404-815-5
6. A. Srinivasan, J. Suresh, “Cloud Computing: A Practical Approach for Learning and Implementation”, *Pearson*, (2014), ISBN 978-81-317-7651-3

Online/Web/Other References

1. <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-291r2.pdf>
[Accessed on 9th December 2025]

2. <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>
[Accessed on 9th December 2025]
3. <https://docs.docker.com> [Accessed on 9th December 2025]
4. <https://www.bmc.com/blogs/devops-basics-introduction/> [Accessed on 9th December 2025]
5. <http://searchdatacenter.techtarget.com/definition/data-center> [Accessed on 9th December 2025]
6. http://www.sapdatacenter.com/article/data_center_functionality/ [Accessed on 9th December 2025]
7. <https://www.salesforce.com> [Accessed on 9th December 2025]

23PCCE603 INFORMATION AND CYBER SECURITY

Teaching Scheme

Lectures: 3 Hours / week

Examination Scheme

In Semester: 50 Marks

End Semester: 50 Marks

Credit: 3

Prerequisites: Computer Networks

Course Objectives

To facilitate the learner to

1. Apply fundamental cybersecurity concepts to analyze various types of cyber threats and vulnerabilities
2. Implement cryptographic techniques to secure data transmission and authentication mechanisms
3. Demonstrate the use of firewalls, VPNs, and intrusion detection systems for network security
4. Utilize digital forensics tools to investigate and mitigate cybercrimes effectively

Course Outcomes

After completion of the course, students will be able to

1. Apply cybersecurity principles to analyze various types of cyberattacks and evaluate security architectures
2. Experiment with cryptographic algorithms like AES, RSA, and hash functions
3. Use network security tools and protocols to safeguard mobile and wireless communications against cyber threats
4. Demonstrate forensic methodologies to investigate cybercrime techniques and propose effective countermeasures

Unit I: Fundamentals of Cybersecurity and Cybercrime

Need and significance of security, Security architectures, Common attacks- DoS, Phishing, SQL Injection, Cross-Site Scripting, Active vs. Passive Attacks, A model for network and internetwork security, cybercrime and information security, cybercriminals, Classification of cybercrimes, Ethical and legal perspectives of cybercrime-Global and Indian ITA 2000.

Unit II: Cryptography and Secure Communication

Cryptography, Classical and Secret Key Cryptography, Cipher basics, Introduction to DES, AES, and IDEA, Block cipher modes of operation, Introduction to public key cryptography, RSA algorithm, Key management basics - Diffie-Hellman, Private/Public Key Distribution.

Unit III: Network and Mobile Security

Network layer security, Firewalls (design principles, types, and working), Concept of VPN,

Tunnelling protocols, Working of IPSEC, Transport layer security -SSL/TLS protocol, Security challenges in mobile and wireless devices, Attacks on mobile/cell phones, security policies and measures for mobile security, Cyber security products and their role.

Unit IV: Cybercrime Techniques, Attack Methods and Defense

Social Engineering, Cyberstalking, Botnet, Proxy servers and Anonymizers, Phishing, Password Cracking, Keyloggers, and Spyware, Viruses, Worms, Trojans, and Backdoors, Steganography, Role in Cybercrime, DoS and DDoS attacks, Deepfake Attacks, Ransomware Evolution.

Unit V: Authentication, Digital Signatures and Organizational Security

Need and significance of message integrity, One-way hash functions- SHA-3, Argon2, Message authentication, Digital signatures: Implementation, algorithms, DSS standards, Digital certificates and X.509-structure and revocation, User authentication mechanisms-Kerberos v4 and v5, Security risks in social media and organizational cyber threats.

Unit VI: Future Trends in Cybersecurity

Digital forensics, Forensic software and hardware tools, Case studies on cybercrime investigations, Emerging threats: AI in cybersecurity, Future trends: Quantum cryptography and security, blockchain for security, Cloud Security, IOT Security and cybersecurity automation.

Text books

1. William Stallings, “Cryptography and Network Security: Principles and Practice”, Pearson, 7th Edition, (2017), ISBN 978-93-325-8522-5.
2. William Stallings and Lawrie Brown, “Computer Security: Principles and Practice”, Pearson, 4th Edition, (2017), ISBN 978-9353438869.
3. Nina Godbole and Sunil Belapure, “Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives”, Wiley India, (2011), ISBN 978-81-265-2179-1.
4. Bill Nelson, Amelia Phillips and Christopher Steuart, “Computer Forensics and Investigations”, Cengage Learning, (2009), 4th Edition, ISBN 978-81-315-1946-2 (or 9781435498839 depending on edition).

Reference Books

1. Atul Kahate, “Cryptography and Network Security”, McGraw Hill Publications, 4th Edition, (2019), ISBN 978-9353163310.
2. Bernard Menezes, “Network Security and Cryptography”, Cengage Learning, (2010), ISBN 978-8131513491.

3. Behrouz A. Forouzan, “Cryptography and Network Security”, McGraw Hill Publications, 4th Edition, (2007), ISBN 978-0-07-287022-0.
4. Bruce Schneier, “Applied Cryptography – Protocols, Algorithms and Source Code in C”, John Wiley & Sons, 2nd Edition, (1996), ISBN 978-1-119-09672-6.
5. Charlie Kaufman, Radia Perlman and Mike Speciner, “Network Security: Private Communication in a Public World”, Pearson Education, 3rd Edition, (2022), ISBN 978-0136643524.

Online/Web/Other References

1. https://onlinecourses.nptel.ac.in/noc25_ee54/preview
2. https://onlinecourses.nptel.ac.in/noc22_cs90/preview
3. https://archive.nptel.ac.in/content/syllabus_pdf/106106129.pdf

23PCCE604 DATA ENGINEERING

Teaching Scheme

Lectures: 3 Hours / week

Examination Scheme

In Semester: 50 Marks

End semester: 50 Marks

Credits: 3

Prerequisite

1. Database Management Systems
2. Artificial Intelligence and Machine Learning
3. Programming Skills in Python

Course Objectives

To facilitate the learners to

1. Understand the core concepts of Data Engineering, Big Data and Analytics
2. Understand storage technologies, storage networking fundamentals and secure storage infrastructure.
3. Get acquainted with exploratory data analysis and tools for data visualization
4. Explore tools like Hadoop, Spark, cloud-based data processing platforms and understand the scalability of big data storage and processing
5. Understand importance and challenges in data governance, data quality, data ethics, and privacy in data science

Course Outcomes

By taking this course, students will be able to

1. Understand the components and functions of intelligent storage technologies and storage infrastructure
2. Apply the principles of data pipelines, data ingestion, ETL, and ELT to design and execute scalable data processing workflows
3. Choose the right data analysis and visualization tools to communicate insights effectively
4. Use big data processing techniques to analyze large-scale datasets using Hadoop or Spark
5. Apply real-time data processing methods and data governance practices to ensure reliable and compliant data operations

Unit I : Data Engineering and Storage Technologies

Introduction to data Engineering, Various roles in data science, types of data, data science workflow Information storage, data storage architecture, Data center: Architecture, core elements, key characteristics, managing data centre; Intelligent Storage Systems, RAID: Array components, Techniques, Levels, Impact on Disk Performance, Front end, Cache, Back end, High-end storage systems; Storage networking technologies, Object-based and Unified storage, Securing the storage infrastructure.

Unit II : Data Ingestion

Identifying potential data sources, Data collection techniques (APIs, Web scraping, Python libraries for data preparation etc.), Introduction to modern data infrastructure, Data pipelines, Architecture, Data Ingestion: Extracting data, Loading data, Data preparation; Orchestrating pipelines: Apache airflow setup, measuring, monitoring and optimizing Data pipeline performance, Handling failures, common patterns - ETL and ELT.

Unit III : Exploratory Data Analysis and Visualization

Exploratory Data Analysis (EDA) fundamentals, significance, steps, software tools, Visual aids for EDA, Comparing EDA with classical analysis, Data Visualization and Storytelling: Dashboard designing with tools like Tableau, Power BI etc., effective data communication strategies, creating Reports and Presentations.

Unit IV: Big Data Analytics

Structured Vs. unstructured data, NoSQL, Types of NoSQL database, Big data: Sources, applications, market drivers; MongoDB : CRUD operations, Aggregation, MapReduce; Distributed and parallel computing: cloud and Big data, Hadoop Eco system, Kafka, Apache Spark: Architecture, components, Hadoop on cloud (AWS, Google cloud, Azure)
Visualizing big data: Need for advanced visualization techniques, big data visualization with Python, Tableau, PowerBI etc.

Unit V : Real time Data Processing and Data Governance

CI/CD for Data Pipelines, Version Control (Git), Testing, Deploying and Monitoring Pipelines in Production Streaming Pipelines: Data Ingestion, Stream Processing, Streaming Analytics frameworks - Apache Flink, Spark Streaming, Message Brokers (Kafka, RabbitMQ)
Data Governance, importance and challenges, Master Data Management, Metadata Management, Data quality Metrics, framework, tools and technologies for Data governance, data Ethics, Bias and privacy in Data Science.
Case Study: End-to-End E-commerce Analytics Pipeline.

Text Books

1. S. Gnanasundaran, Alok Shrivastava, "Information Storage and Management" 2nd Edition, Wiley Publishers, ISBN : 978-81-265-3750-1
2. V. K. Jain , "Data Science and Analytics", Khanna Books Publishing Ltd, ISBN : 978-93-86173-67-6
3. "Big Data, Black Book", DreamTech Press, ISBN : 978-93-5119-931-1
4. Dimitrios Sargiotis, "Data Governance, a Guide", Springer, ISBN : 13: 978-3031672675

Reference Books

1. Jiawei Han, Micheline Kamber, and Jian Pie, "Data Mining: Concepts and Techniques", Elsevier Publishers
2. Chirag Shah, "A Hands-On Introduction To Data Science", Cambridge University Press

3. “Data Pipelines Pocket Reference, Moving and Processing data for Analytics” by O’Really
4. Suresh Kumar Mukhiya, “Hands on EDA with Python”, Packt Publishing
5. “Data Science and Big Data Analytics”, Wiley Publications
6. James Densmore, “Data Pipelines Pocket Reference, Moving and Processing data for Analytics”, O’Really

Online/Web/Other References

1. An Introduction to Statistical Learning by Gareth James (<https://www.ime.unicamp.br/~dias/Intoduction%20to%20Statistical%20Learning.pdf>)
2. Introducing Data Science by Davy Ciele, Manning Publications
3. Introducing Data Science [PDF]
4. Handbook for visualizing : a handbook for data driven design by Andy Kirk
5. An introduction to data Science (<https://docs.google.com/file/d/0B6iefdnF22XQeVZDSkxjZ0Z5VUE/edit?pli=1>)
6. Hadoop Tutorial (https://www.tutorialspoint.com/hadoop/hadoop_tutorial.pdf?utm_source=7_&utm_medium=affiliate&utm_content=5f34cd37cdf1050001b09537&utm_campaign=Admitad&utm_term=761c575424fc4a6b48d02f72157eb578)
7. Learning with Python; How to think like a computer scientist (<http://openbookproject.net/thinkcs/python/english3e/>)
8. Python for everybody (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf)

23PECE601A DevOps FUNDAMENTALS

Teaching Scheme

Lectures:3 Hours / week

Examination Scheme

In Semester: 50 Marks

End Semester: 50 Marks

Credits: 3

Prerequisites: Software Engineering and Design Architectures

Course Objectives

To facilitate the learner to

1. Understand the need for the DevOps as a state-of-art software engineering practice
2. Learn the basic concepts related to DevOps
3. Get acquainted with the various tools which are used in different phases of the DevOps model
4. Get exposure to emerging trends in software development related to DevOps

Course Outcomes

After completion of the course, students will be able to

1. Apply the fundamental concepts and emerging trends of DevOps to software development
2. Examine the relevance of DevOps for real life application development
3. Make use of the various tools of continuous integration, continuous delivery, continuous testing and monitoring in DevOps
4. Make use of the various deployment platforms as part of DevOps lifecycle

Unit I: DevOps Introduction

Overview, Features, Components, Benefits , Business need for DevOps, Using DevOps to solve new challenges like enabling mobile applications, Managing multi tier applications, DevOps lifecycle, Develop, Code, Build, Test, Deploy, Use cases, Continuous improvement, Release planning, Continuous integration, Continuous delivery, Continuous testing, Continuous monitoring and feedback, DevOps tools.

Unit II: Application Development

Automating SDLC and DevOps, DevOps team, Serverless computing, Monolithic to microservices, Microservices characteristics, Use cases, Architecture, Services design.

Unit III: Continuous Integration Continuous Delivery (CI/CD) Pipeline

CI/CD pipeline, Source code repository, Version control and source code management, GitHub, Git commands, Creating automated build, Automated build frameworks like Maven, Ant for Java, Automated configuration and automated deployment, Configuration management with tools like Puppet, Chef and Ansible, Continuous integration with Jenkins.

Unit IV: Continuous testing and Continuous monitoring

DevOps and testing, Release testing, Testing in development, Testing in production, Testing and bug tracking frameworks such as JUnit, Selenium, TestNG and JIRA. Monitoring process, Monitoring principles and practices, Monitoring metrics, monitoring benefits. Continuous monitoring and logging with tools like Nagios, Grafana, ELK stack.

Unit V: Deployment Platforms

Containerization and DevOps, Benefits of containers in enabling DevOps workflow, Containerization Tools, Docker for DevOps, Docker runtime container, Docker installation, Commands, Docker use cases, Container Orchestration, Kubernetes.

Unit VI: DevOps Applications and Trends

Cloud's benefit to DevOps, Web Applications on Cloud Platform, Automation of infrastructure, Infrastructure as code (IaC) software tool Terraform, DevSecOps, Low code solutions, MLOps, AIOps.

Text Books

1. Sanjeev Sharma and Bernie Coyne, 'DevOps for Dummies', *IBM Limited Edition*, John Wiley and Sons, Inc., ISBN- 978-126553495, (2017).
2. Viktor Farcic, 'The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices', *CreateSpace Independent Pub*, 978-1523917440(2016).
3. Katrina Clokie, 'A Practical Guide to Testing in DevOps', *Leanpub*, ISBN-9781727132175 (2017).

Reference Books

1. Bass, L., Weber, I.M., Zhu, L., 'DevOps: a software architect's perspective', Pearson Education, ISBN: 9789332570375, (2016).
2. Davis J., Daniels K., 'Effective DevOps: Building a Culture of Collaboration, Affinity and Tooling at Scale', O'Reilly, ISBN- 9789352133765, (2018).
3. Farooqui S. M., 'Enterprise DevOps Framework: Transforming IT Operations', CA Press / Apress, ISBN- 9781484240618, (2019).
4. Sanjeev Sharma, 'The DevOps Adoption Playbook: A Guide to Adopting DevOps in a Multi-Speed IT Enterprise', Wiley, ISBN- 9788126569083, (2017).

Online/Web/Other References

1. <https://devops.com/>
2. <https://docs.docker.com>
3. <https://www.bmc.com/blogs/devops-basics-introduction/>
4. <https://www.ibm.com/in-en/cloud/devops>
5. <https://aws.amazon.com/devops/what-is-devops/>

23PECE601B DEEP LEARNING

Teaching Scheme

Lectures: 3 Hours / week

Examination Scheme

In Semester: 50 Marks

End Semester: 50 Marks

Credits: 3

Course Objectives

To facilitate the learner to

1. Understand concepts of Artificial Neural Network and Deep Neural Network
2. Understand Artificial Neural Network models
3. Understand Deep Neural Network models and architectures
4. Select appropriate Neural Network model for various applications

Course Outcomes

After completion of the course, students will be able to

1. Make use of Artificial Neural Network and Deep Neural Network concepts for problem-solving
2. Construct an Artificial Neural Network model for given data
3. Identify suitable Deep Neural Network architectures for real-world applications
4. Understand advanced Deep Learning concepts for real-world applications

Unit I: Artificial Neural Networks

Biological Neuron, McCulloch-Pitts neuron, Artificial Neural Networks (ANN), The Perceptron, Activation and Loss Functions, linearly separable and inseparable data, Multilayer Perceptron (MLP), Types of Activation Functions, Loss Functions, Confusion Matrix for Categorical Data, Training a Neural Network with Backpropagation, Practical Issues in Neural Network Training, The Problem of Overfitting, Regularization, EarlyStopping.

Unit II: Convolution Neural Network

Introduction to Deep Learning, Deep Learning Applications, Popular Open Source Libraries, The Basic Structure of a Convolutional Network, Convolution Layer, Padding, Strides, The ReLU, Pooling, Fully Connected Layers, Training a Convolutional Network, The Vanishing and Exploding Gradient Problem, Dataset Augmentation.

Unit III: Advanced Convolution Neural Network

Architectures: LeNet-5, AlexNet, VGG Net, ResNet, Applications: Semantic Segmentation, Instance Segmentation, Object Detection, Learnable parameters, Parameter Sharing and Sparse Interaction, Case Study for Image Classification.

Unit IV: Recurrent Neural Networks and Word Embeddings

Working with Text Data, Using Word Embeddings and Word2Vec, Recurrent Neural Network (RNN) Architecture, Backpropagation Through Time, Applications of Recurrent Neural Networks, The Challenges of Training Recurrent Networks, Long Short-Term Memory (LSTM) Networks, Gated Recurrent Units (GRUs), Attention Mechanisms in RNNs, Case Study: Sentiment Classification

Unit V: Autoencoders and Advanced Deep Learning

Autoencoders Architecture and Applications, Deep Autoencoders, Generative Adversarial Networks (GANs), Introduction to Large Language Models, Transformer Architecture, Deep reinforcement Learning, Applications of DNN, Transformer and GAN.

Text Books

1. Charu C. Aggarwal, "Neural Networks and Deep Learning", Springer International Publishing, Second Edition, (2023), ISBN 978-3-319-94462-3 ISBN 978-3-319-94463-0
2. Nikhil Baduma, Nicholas Locascio, "Fundamentals of Deep Learning: Designing Next Generation Intelligence Algorithms", O'Reilly Publication, (2017), ISBN 10: 9352135601, ISBN 13: 978- 9352135608,
3. Christopher M. Bishop, "Deep Learning: Foundations and Concepts", Hugh Bishop, Springer, (2024), e-book (free) Deep Learning - Foundations and Concepts

References Books

1. Ian Goodfellow and Yoshua Bengio and Aaron Courville, "Deep Learning", MIT Press Ltd., (2017), ISBN: 9780262035613
2. FRANÇOIS CHOLLET, "Deep Learning with Python", Manning Publications Co., (2017), ISBN 9781617294433,
3. Josh Patterson and Adam Gibson, "Deep Learning – A Practitioner's approach", O'Reilly Publication, 1st edition, (2017), ISBN : 9789352136049
4. Shekhar Khandelwal, "Deep Learning for Data Architects", BPB Publication, (2023), ISBN: 9789355515391 eISBN: 9789355515292

Online/Web/Other References

1. URL: <https://d2l.ai>
2. <https://www.deeplearningbook.org>
3. <https://jalammar.github.io/illustrated-transformer/>

23PECE601C SOFTWARE TESTING AND QUALITY ASSURANCE

Teaching Scheme

Lectures: 3 Hours / Week

Examination Scheme

In Semester: 50 marks

End Semester: 50 marks

Credits: 3

Prerequisites: Software Engineering and Design Architecture

Course Objectives

To facilitate the learner to

1. Develop familiarity with the fundamental concepts and the process of software testing
2. Gain comprehensive knowledge about various software testing techniques and methods
3. Study various software testing strategies
4. Get exposure to the quality assurance process and its role in software development
5. Get acquainted with the agile testing methods, the recent trends and tools used for testing different types of applications

Course Outcomes

After completion of the course, students will be able to

1. Examine software testing techniques and strategies in the context of various problem areas
2. Make use of the concepts of testing metrics and quality assurance for software systems
3. Utilize agile testing strategies for testing various types of applications
4. Apply the automated testing and emerging trends of software testing to real life scenarios

Unit I: Software Testing Fundamentals and Techniques

Need of testing, Basic testing terminologies, Testing Principles, Goals, Software Testing Life Cycle, Defects, Defect management, Test Plan.

Static testing, Dynamic testing, Black box testing, White-Box testing, Grey box testing, Black box testing Methods - Requirements based testing, Positive and negative testing, Decision table / Cause effect graphing, State based testing, Domain testing, Graph based testing,

White box testing Methods - Structural Testing, Test adequacy criteria, Dataflow testing, Loop testing, Mutation testing, Penetration testing, Challenges in testing, Case studies of Black-Box and White-Box testing.

Unit II: Testing Levels and Strategies of Testing

Testing Levels - Unit, Integration, System, Acceptance testing

Testing Strategies - Usability testing, Sanity/Smoke testing, Regression testing, Scenario testing, Adhoc testing, Functional testing, Performance testing, Load testing, Stress testing, Security testing, Alpha-Beta testing, Recovery testing, Configuration testing, Compatibility testing, I18N testing, L10N testing, Content testing, User interface testing, Component-level testing, Case studies.

Unit III: Testing Metrics and Quality Assurance

Testing Metrics and measurements, Types of metrics – Project, Progress, Productivity, Software quality, Quality control and assurance, Quality factors, Software Quality Assurance (SQA) Model - Six Sigma, Ishikawa's Seven Basic Tools, Quality Standards - ISO, CMMI, TMMi.

Unit IV: Agile Testing

Introduction, Agile testing Principles, Agile testing methodologies - Behavior Driven Development (BDD), Exploratory testing, Acceptance Test Driven Development (ATDD), User Story testing, Benefits and challenges.

Unit V: Automated Testing

Need for Automation, Keyword driven automation, Data driven automation, Manual testing versus Automated testing, Automated Testing Tools and frameworks such as Selenium, JUnit, JIRA, JMeter, Postman.

Unit VI: Recent Trends in Testing

Shift-Left and Shift-Right Testing, Cloud-Based Testing, Model-based testing, API and Microservices Testing, Autonomous testing, Continuous Integration and Continuous Testing (CICT), Role of AIML in testing – Test data generation, Predictive analytics for defect prevention and risk assessment, Test case generation, Prioritization, Optimization, Test execution, Result analysis, Continuous learning, Adaptability, Intelligent model generation.

Text Books

1. Srinivasan Desikan, Gopaldaswamy Ramesh, 'Software Testing Principles and Practices', *Pearson Education*, ISBN 81-7758-121-X (2013).
2. Nageshwar Rao Pusuluri, 'Software Testing Concepts and Tools', *Dreamtech press*, ISBN 81-7722-712-2 (2008).
3. Lisa Crispin, Janet Gregory, 'Agile Testing: A Practical Guide For Testers And Agile Teams', *Addison-Wesley*, ISBN 9788131730683 (2012).
4. Srinivasa Rao Bittla, 'AI-Driven Software Testing', *Apress*, ISBN 979-8-8688-1828-8 (2025).

Reference Books

1. Iien Burnstein, 'Practical Software Testing', *Springer (India) private limited* (2005).
2. Ron Patton, 'Software Testing', *Pearson Education*, ISBN-13: 978-0-672-32798-8 (Second Edition) (2013).
3. Stephen H Kan, 'Metric and Model in Software Quality Engineering', *Pearson Education* ISBN 81-297-0175-8 (Second Edition) (2006).
4. William Perry, 'Effective Methods for Software Testing', *Wiley Publication*, ISBN 81-265-0893-0 (Third Edition) (2006).
5. Dr. K.V.K.K. Prasad, 'Software Testing Tools', *Dreamtech Press* ISBN: 10:81-7722-532-4 (2008).
6. Naresh Chauhan, 'Software Testing Principles and Practices', *Oxford University Press*, ISBN 0-19-806184-6 (2011).

Online/Web/Other References

1. <http://www.seleniumeasy.com/selenium-tutorials>
2. <https://junit.org>
3. <https://www.postman.com/>
4. <https://jmeter.apache.org/>
5. <https://www.betabreakers.com/case-studies/>
6. https://onlinecourses.nptel.ac.in/noc22_cs61
7. <https://nptel.ac.in/courses/106105150>

23PECE601D DISTRIBUTED SYSTEMS

Teaching Scheme

Lecture: 3 Hours / week

Examination Scheme

In semester: 50 marks

End semester: 50 marks

Credits: 3

Prerequisites

1. Operating Systems
2. Microprocessor and Microcontroller

Course Objectives

To facilitate the learner to

1. Learn fundamental concepts of Distributed Systems
2. Learn communication methodology and synchronization techniques in distributed systems
3. Learn replication techniques and consistency models in Distributed Systems for fault tolerant systems
4. Understand the various issues related to distributed databases and transactions
5. Explore trends and applications of Distributed Systems

Course Outcomes

After completion of the course, students will be able to

1. Make use of the concepts of distributed system types and architectures for real time applications
2. Solve coordination problems in distributed environments by applying suitable communication mechanisms and synchronization algorithms
3. Utilize distributed memory management techniques and transaction handling methods to address consistency and concurrency issues
4. Apply different methods to solve issues related to distributed memory and transactions
5. Understand trends and applications of distributed systems

UNIT I: Introduction to Distributed System

Characteristics, System design issues, Types of Distributed Systems, Architecture, Distributed Operating System, Interprocess Communication- message passing, RMI, RPC.

UNIT II: Synchronization

Physical Clocks, Clock Synchronization Algorithms, Logical Clocks–Lamport's Logical clocks, Vector Clocks, Election Algorithms, Global State and Snapshot Recording Algorithms, Mutual

Exclusion Issues, Mutual Exclusion Algorithms- Centralized, Distributed Algorithm, Token-Ring Algorithm, Decentralized Algorithm.

UNIT III: Replication and Fault Tolerance

Reasons for replication, Replica management, Content replication and placement, Content distribution, Managing replicated objects, Consistency protocols, Failure models, Agreement Protocol, Recovery Case Study: Caching and replication in web.

UNIT IV: Distributed Transaction Processing and Memory Management

Distributed Transactions, Flat and Nested distributed transactions, Atomic commit protocols, Concurrency Control in Distributed Transactions, Transaction Recovery, Distributed Deadlocks, Distributed Shared Memory(DSM) concepts, Design and Implementation issues in DSM.

UNIT V: Distributed Algorithms

Distributed consensus, Paxos algorithm, Failure detectors, Termination Detection in Distributed System, Self-Stabilization, Distributed Randomized Algorithms, Distributed Hash Table (DHT) and Peer to Peer (P2P) Computing.

UNIT VI: Trends and Emerging Technologies in Distributed Systems

Advanced Computing Trends, Emerging Distributed Frameworks and Technologies, Trends in Distributed Systems, Ubiquitous Edge Computing, Blockchain, Cloud Computing, Serverless Computing, Security techniques in distributed systems.

Text Books

1. George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair, “Distributed Systems – Concept and Design”, Pearson, fifth Edition,(2017) ISBN – 978-13-214301-1.
2. Tanenbaum and Van Steen, “Distributed Systems: Principles and Paradigms”, Pearson, third edition (2017) ISBN:978-0-13-088893-8

Reference Books

1. Dr. Sunita Mahajan , Seema Shah, “Distributed Computing”, Oxford University Press, Second Edition, ISBN No. 978-01-980-9348-0
2. Pradeep K Sinha, “Distributed Operating System”, PHI, (1996) ISBN – 978-81-203-1380-4.
3. A.D. Kshemkalyani, M. Singhal, “Distributed Computing: Principles, Algorithms, and Systems”, Cambridge University Press, (2011) ISBN: 9780521189842
4. Nancy Lynch, “Distributed Algorithms” , Morgan Kaufmann Publishers (1196) ISBN-13:978-1-55860- 348-6

5. A. Silberschatz, H. Korth, and S. Sudarshan, "Database Systems Concepts", McGraw-Hill, New York, 7 edition, (2010)

Online/Web/Other References

1. Prof. Rajiv Misra, Distributed System, <https://nptel.ac.in/courses/106106168>
2. Prof. Ananthanarayana V.S., Distributed Systems, <https://nptel.ac.in/courses/106106107>
3. Prof. Rajiv Misra, Cloud computing and Distributed System <https://nptel.ac.in/courses/106104182>

23MmCE601 ENTERPRISE BANKING AND INSURANCE

Teaching Scheme

Lectures : 2 Hours / week

Examination Scheme

In Semester: 25 Marks

End Semester: 25 Marks

Credits: 2

Course Objectives

To facilitate the learner to

1. Understand the basic concepts, structures and risk management in banking
2. Become familiar with the concept of insurance and its categories
3. Introduce insurance product development basics and role of actuary
4. Become acquainted with the trends in banking

Course Outcomes

After completion of the course, students will be able to

1. Outline the banking basics, its types and risk management for an enterprise
2. Select the insurance based on the needs of the customer
3. Identify the steps in the insurance product development process and underwriting for a small business
4. Understand various trends in banking and technology

Unit I: Banking Basics

Indian Banking System - Evolution, Reserve Bank of India and its role, Structure of Banks in India, role of banks in economic development, Financial inclusion schemes in India, Traditional vs modern banking, Banking Types: Retail, Private, Corporate, Investment, International; banking risks and risks management.

Unit II: Basics in Insurance and its Types

Introduction, Evolution, Principles, Purpose and need of Insurance in India; Insurance as a social security tool, Insurance: Markets, Customers, Contracts; Broad categories of Insurance - Life, General, Health, Specialized, Indemnity-based and Benefit-based Insurance, Role of Insurance in Economic Development.

Unit III: Functions in Insurance and Compliance

Appointment & Role of Appointed Actuary, Product Pricing, Reserving, Product Review, Actuarial Valuations, Review of Financial Condition / Economic condition, Product Design & Filings, File & Use and Use & File, Product Filing Guidelines, Underwriting for Risk selection,

Risk Tolerance, Rating Pricing, Loading & Discounts.

Unit IV: FinTech

Banking as a service (BaaS), Use of Artificial intelligence and Blockchain in Banking, Digital banking, Open banking, Case study on Oracle banking, Case studies

Text Books

1. Barbara Casu, Claudia Girardone, Philip Molyneux, "Introduction To Banking", 3rd Edition, Pearson Publication (2021)
2. The Institute of Company Secretaries of India, "Banking and Insurance-Laws and Practice"

Reference Books

1. School of Distance Education, University of Calicut, "Banking and Insurance"
2. Oracle® Banking Supply Chain Finance, Release 14.7.2.0.0, F89725-01 November 2023

Online/Web/Other References

1. <https://www.capco.com/intelligence/capco-intelligence/top-5-banking-trends-for-india-2025> (last access: 05/12/2025)
2. https://onlinecourses.swayam2.ac.in/imb24_mg50/preview (last access: 05/12/2025)

23VSECCE601L PROGRAMMING SKILLS DEVELOPMENT LABORATORY II

Teaching Scheme

Lectures: 4 Hours / week

Examination Scheme

In Semester: 25 Marks

End semester: 25 Marks

Credits: 2

Prerequisite

1. Programming Skills in Java Language
2. Data Structures

Course Objectives

To facilitate the learner to

1. Explore the usage of mobile development tools
2. Learn the process of development of mobile applications
3. Create data-driven mobile applications
4. Create a mini project using all the concepts

Course Outcomes

After completion of the course, students will be able to

1. Select suitable configuration parameters, components, and API's, libraries for mobile application development
2. Design a mobile application using suitable widgets
3. Develop mobile application with database connectivity
4. Build a suitable mobile application for a real life scenario

Preamble

The lab would be for understanding the syntax and semantics of Android/Flutter framework and implementation of some real-world simple applications. Assignment statements are in brief and should be implemented using the Android/Flutter framework. Motivation here is that students should be able to learn App development and also should be able to analyze problems and select suitable built-in tools/API. Faculty members are encouraged to expand problem statements with variations. Assignments can be framed and expanded in such a way that it explores concepts, constructs and design of simple applications. Students will be encouraged to solve open ended problems in different domains. Faculty will appropriately adopt assignments on similar lines as the examples shown here.

Group A assignments are for learning the basics of Android/Flutter framework. Group B assignments are for learning advanced features to the App. Group C assignment is open ended application development.

Suggestive List of Assignments

Assignment 0 : Download, install and configure android/flutter framework.

Group A: (Mandatory)

1. Develop a mobile application to perform scientific mathematical operations using UI Widgets.
2. Develop a form that uses a dropdown list.
3. Develop an application with multiple screens and navigation options.
4. Develop an application that makes use of Alert Dialog to navigate between different web pages.
5. Develop a mobile application that makes use of cloud databases (e.g. Firebase).

Group B: (Perform any Three)

1. Develop a mobile application to generate a chatbot app.
2. Develop a mobile application for multimedia application to simulate audio and video features.
3. Develop a mobile application that implements Multithreading.
4. Develop a mobile application that creates an alert upon receiving a message.
5. Develop a mobile application that uses a time picker.
6. Develop a mobile application that makes use of URI (Uniform Resource Identifier), XML Pullparser like RSS (Really Simple Syndication) feed.
7. Develop a mobile application for image transformations like Translation, Scaling and Rotation.
8. Develop an application that reads/writes data to-from the SD card.
9. Develop an application for sending email/SMS.
10. Develop a sensor based application using Motion sensors/Position sensors/Environmental Sensor.

Group C: (Mandatory)

Consider a scenario of your choice. Design and develop a mobile application for the selected scenario.

For the development of this mobile application, choose the appropriate flutter framework constructs like components, widgets and layout.

Build the APK file of the developed mobile application and deploy/publish it to the play store or distribute it directly.

23PCCE601L FULL STACK DEVELOPMENT LABORATORY

Teaching Scheme

Practical: 2 Hours / week

Examination Scheme

In Semester: 25 Marks

End Semester: 25 Marks

Credit: 1

Course Objectives

To facilitate the learners to

1. Understand the Installation and Configuration setting related aspects of web server, integrated development environments and various frameworks
2. Understand the role of various technologies used for real-life application development
3. Get exposure to full stack development in Java which includes client side and server side technologies, web services and database development technologies
4. Gain practical knowledge about the various client side and Java server side technologies for application development in practice

Course Outcomes

By taking this course, the learner will be able to

1. Make use of suitable client side Java technologies for web application development
2. Experiment with various Java server side technologies like Java Servlets, Java Server Pages, Web services and Java Persistence API (JPA)
3. Make use of Java Sockets library and Java Remote Method Invocation(RMI) framework for the development of sample client-server applications
4. Build a sample web application using suitable technologies at various tiers

Preamble

Development of web applications need technologies at various levels, which play different roles in the overall web architecture. The intent of Java Full Stack Technologies Laboratory is to enable the understanding of the role of various technologies in full stack development and implementation of some real world application scenarios using these technologies. Assignment statements are in brief and should be implemented with Java web technologies. Motivation here is that students should be able to develop the user interface, business logic and the database programming parts of a typical web application, using the APIs/libraries provided by various client side and server side Java technologies. Faculty members are encouraged to expand problem statements with variations. Assignments can be framed and expanded in such a way that it explores concepts, logic of solution and simple application. Students will be encouraged to build solutions for real world business scenarios in different domains, to fulfil the end-user requirements. Faculty will appropriately adopt assignments on similar lines as the examples shown here. Group A assignments are on applying various client side technologies and basic server side technologies. Group B assignments are on exploring the use of technologies like Web services and Hibernate. Group C assignment is on the development of sample web applications.

Suggestive List of Assignments

Group A: (Mandatory)

1. Develop dynamic and interactive web client using Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript technologies. Make use of these technologies to develop suitable web forms, layout and to perform validation of form data, for this web client. Sample application scenario: Consider that a student needs to register for an online course portal. For this scenario, develop an HTML form for "Course Registration", make use of CSS for layout design of this form and perform validation on various fields of this form using JavaScript.
2. Develop dynamic and interactive web client using Extensible Markup Language (XML) and Asynchronous JavaScript And XML (AJAX) technologies, to enable rich user experience. Sample application scenario for AJAX: Consider a web form for an administrator of an "Online Shopping Application". An administrator can select the name of a customer from the drop down list box on the web form. Then on the same page, the details of the customer such as shipping address should get displayed.
3. Develop a dynamic and interactive web client using ReactJS as a client side library. Make use of various features of ReactJS such as components, APIs etc.
4. Implement a simple client-server application like echo server or chat server using Java Sockets. Make use of multithreading in Java for handling the requests from multiple clients.
5. Implement an application using Java Remote Method Invocation (RMI) to understand distributed application environment. The remote object accesses the database using JDBC.
6. Implement a sample web application scenario using Java Servlets, Java Server Pages and Java Beans as the server side dynamic content generation technologies. Make use of Model View and Controller (MVC) architecture for this implementation and also show the appropriate usage of the various capabilities of these technologies such as session tracking, tag library, implicit objects, directives. Sample application scenario: Consider a simple web form where you give a Student Roll number and get back Student Profile details from the database. Make use of MVC architecture, based on Java Servlet, JSP and Java Bean to implement this web based scenario.

Group B: (Any One)

1. Create a JSP based Web Application using Spring Boot framework.
2. For a sample application scenario, implement and consume the suitable web services using Service Oriented Architecture Protocol (SOAP).
3. For a sample application scenario, implement and consume the suitable web services using Representational State Transfer (REST) protocol.
4. Make use of JPA with Hibernate framework for performing the create, retrieve, update and delete (CRUD) operations on the backend database. Sample application scenario: Consider "Course Information Management" as a typical Database Application. This application may have database tables like Courses, Participants etc. Make use of JPA

with Hibernate framework to access the data from the Courses table in this above application.

Group C: (Mandatory)

Design and develop a web application like online cab booking, online food ordering application, online tours and travel portal. For the development of this application, choose the appropriate technologies for the client side aspects, server side business logic and database development.

23PCCE602L CLOUD COMPUTING LABORATORY

Teaching Scheme

Practical: 2 Hours / week

Examination Scheme

In semester: 25 Marks

End semester: 25 Marks

Credits: 1

Course Objectives

To facilitate the learners to

1. Explore the underlying principles of Infrastructure-as-a-Service (IaaS), virtualization and containers
2. Understand the use of the Hadoop ecosystem
3. Get exposure to the use of cloud Application Programming Interfaces (APIs) for developing sample application(s)
4. Study different cloud platforms and tools for various cloud service models

Course Outcomes

By taking this course, the learner will be able to

1. Make use of hypervisor and container-based virtualization.
2. Experiment with the Hadoop ecosystem by implementing sample programs for Hive/HDFS/Map-Reduce.
3. Make use of CloudSim framework for understanding cloud computing infrastructure and services.
4. Examine the use of different cloud platforms and tools/APIs for various cloud service models.

Preamble

The intent of Cloud Computing Laboratory is to enable the understanding and implementation of the basic concepts of Cloud Computing. Assignment statements are in brief and can be implemented with Java/Python programming language. Motivation here is that students should be able to experiment with different aspects of IaaS, PaaS and SaaS using various APIs/libraries. Faculty members are encouraged to expand problem statements with variations. Assignments can be framed and expanded in such a way that it explores concepts, logic of solution and simple application. Students will be encouraged to explore different cloud platforms and tools. Faculty will appropriately adopt assignments on similar lines as the examples shown here. The basic and the next level experimentation with CloudSim, Docker container, virtualization and Hadoop ecosystem is covered by the assignments in Group A and those in Group B, respectively. Group B assignments are also on exploring the various cloud APIs. Group C assignments are on exploring the various cloud platforms.

Suggestive List of Assignments

Group A: (Mandatory)

1. Explore the CloudSim platform for cloud modelling. For example: Create a data centre with one host and run one cloudlet on it using CloudSim.
2. Demonstrate the use of Docker containers by exploring its related commands. Also, show the use of Fedora/Ubuntu images over the Docker engine.
3. Demonstrate the use of MySQL/Tomcat/MongoDB image over the Docker engine.
4. Demonstrate the use of Hive query language (HQL) to process the data using the Hadoop ecosystem.
5. Create a virtual machine using Kernel Virtual Machine (KVM) and explore commands for virtualization.

Group B: (Any Three)

1. Experiment with the CloudSim platform for modelling and simulation of cloud infrastructure. For example: Create and configure the data centre and user base to show response time, request servicing time and data centre loading.
2. Frame Python scripts to perform operations (for e.g. start/pause/stop) on the Virtual Machine using Libvirt and Operating System (OS) calls for virtualization.
3. Build the Docker image from a Docker file and demonstrate the use of it over the Docker engine.
4. Using the Hadoop ecosystem, implement Map-Reduce word count program on a single node cluster for the given sample data.
5. Using the Hadoop ecosystem, implement a Map-Reduce program for the given log file data.
6. Explore and configure the Xen/VirtualBox/VMware hypervisor.
7. Execute Hadoop Distributed File System (HDFS) commands on Hadoop ecosystem.
8. Install Google App Engine. Create hello world application and other simple web applications using Python/Java.
9. Explore the use of API for cloud storage applications (for e.g. DropBox API) with the Linux command line interface and Python script.
10. Create an application using Force.com API.
11. For a sample application, implement and consume web service using social networking APIs with Simple Object Access Protocol (SOAP).
12. For a sample application, implement and consume web service using cloud APIs with REpresentational State Transfer (REST).

Group C: (Any One)

1. Explore an open source Cloud platform with respect to architecture, installation and configuration. Also, demonstrate its functionality through practical use cases.
2. Explore the use of different cloud platforms and tools such as Google App Engine (GAE), Amazon Platform Services, Microsoft Azure services, Openstack, Rackspace, Salesforce.com and Kubernetes.

23PCCE604L DATA ENGINEERING LABORATORY

Teaching Scheme

Practical : 2 Hours /week

Examination Scheme

In semester: 25 Marks

End semester: 25 Marks

Credits: 1

Prerequisite :

1. Database Management Systems
2. Artificial Intelligence and Machine Learning
3. Programming Skills in Python

Course Objectives

To facilitate the learners to

1. Understand and practice the core concepts of data science, Big Data, and analytics
2. Explore various data sources, data preparation, Data Pipelines and Orchestrating Pipelines
3. Explore tools and technologies like Hadoop, Spark, MongoDB, cloud-based data processing platforms and understand the scalability of big data storage and processing
4. Get acquainted with Exploratory Data analysis and tools for Data Visualization and Storytelling

Course Outcomes

By taking this course, students will be able to

1. Apply the Data Science and Big Data Analytics techniques for effective data-driven decision-making
2. Make use of tools viz. Tableau / PowerBI / Spark / Talend to analyze data
3. Apply appropriate data visualization techniques using Python or other visualization tools to present insights effectively
4. Build an end-to-end data processing workflow using Python or PySpark, encompassing extraction, transformation, and loading

Preamble

Data engineering is a fundamental discipline in this Big Data era, which enables organizations to make data-driven decisions. This laboratory provides practical exposure to building robust, scalable, and efficient data infrastructure. Students will learn to implement Extract, Transform, Load (ETL) pipelines, work with both structured (SQL) and unstructured (NoSQL) data storage solutions, and manage data workflows using modern, industry-standard tools.

The core objective of this lab is to familiarize students with the entire data engineering lifecycle. Specifically, the laboratory aims to: Understand and practice the core concepts of data science,

Big Data, and analytics, explore various data sources, data preparation, data pipelines and orchestrating Pipelines, and explore tools and technologies like Hadoop, Spark, MongoDB, cloud-based data processing platforms and understand the scalability of big data storage and processing. The lab also aims students to get acquainted with Exploratory Data analysis and tools for Data Visualization and Storytelling.

Suggestive list of assignments

Group A Assignments (All)

- 0 Install data analytics tools like Tableau, powerBI etc. and explore
1. Choose a dataset from UCI Machine Learning repository (e.g. Cleveland) and
 - A. Create basic visualizations like bar charts, pie charts, and scatter plots to explore the data.
 - B. Compute and display summary statistics for each feature available in the dataset. (eg. minimum, maximum, mean, range, standard deviation, variance and percentiles). Use a bar-graph to demonstrate your results.
 - C. Create a histogram for each feature in the dataset to illustrate the feature distributions. Plot each histogram.
 - D. Create a box-plot for each feature in the dataset. Compare distributions and identify outliers.
2. Consider a MongoDB collection and perform the CRUD operations. Use the aggregate() and MapReduce() functions to extract meaningful insights from a dataset.
3. Use a large public open source image dataset (e.g., CIFAR-10, ImageNet subset, Google Open Images), Load it into a distributed file system such as HDFS & use Spark framework to -
 - A. Resize images
 - B. Convert formats (PNG → JPEG)
 - C. Extract basic metadata (dimensions, color channels, histograms) and visualize.

Group B Assignments (Any 4)

1. Write a PySpark application that imports a text-based dataset from any of the file formats (such as csv, Excel, word etc.) which will identify each distinct word, and computes how often each item appeared in the dataset. Export the resulting frequency distribution to a text file.
2. Develop a Spark / Python application to perform data aggregation and filtering on a sample dataset
3. Use Spark / Panda / Polar DataFrames and SQL to perform complex queries and transformations on sample data
4. Build a complete ETL workflow in python/PySpark that takes data from different sources, cleans it, applies required transformation rules(data conversion, functional requirements change), and then loads it step-by-step into a data warehouse.
5. Implement Simple Naïve Bayes classification algorithm (or any other classical Machine Learning algorithm) using Python on a sample dataset. Compute Confusion matrix to

- find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset. Perform analysis on various parameters using a visualisation tool.
6. Take any dataset from UCI repository (like air quality dataset) and perform regression analysis on it. Demonstrate your results using appropriate visualization techniques for numerical and categorical features (e.g. histogram, scatter plot, heat map, box plot).
 7. A retail company wants to analyze sales data from multiple sources (web, mobile, and stores) in real time.
 1. Ingestion: Collect data from CSV, API, and SQL databases
 2. Storage: Store raw data on the cloud server
 3. Processing: Use Spark for batch processing
 4. Pipeline: Build an automated ETL pipeline with Apache Airflow.
 5. Analytics: Query and visualize insights using SQL and a dashboard (Tableau).

Group C Assignments (Any one)

1. Develop a Spark application using Spark Streaming on the live data stream such as social media feeds, financial data feeds etc., and analyze the posts for their sentiments.
2. Simulate temperature sensor data using a Kafka Producer, and consume the data using Spark Streaming.
3. Patient Risk Healthcare Dashboard : Analyze a massive dataset of patients in a particular hospital and, based on their information, predict and infer the risk of their health. Students should then integrate all this analysis into Tableau for easy consumption of the end-users. The dataset may have many variables for which a thorough analysis has to be done to get details about the patients at risk.
4. A Flight Price Analysis Dashboard : Analyze a dataset of flight-related information. The main objective is to consider different factors of a flight and infer accurate trends for flight prices. The dataset may have many variables that involve date and time data types as well.