`

BTech. CSE V Sem Syllabus
CSE/AIML- 501 Subject: Agile Methodology

COURSE OUTCOMES:

After Completing the course student should be able to

CO1	Understand Agile software development principles and practices
CO2	Apply Scrum values and practices to relevant software application
CO3	Analyse product management agile approach for real life problem
CO4	Assess software for agile requirements and architecture
CO5	Design risk management plan with agile approach

UNIT I AGILE SOFTWARE DEVELOPMENT

Basics and Fundamentals of Agile Process Methods, Values of Agile, Principles of Agile, stakeholders, Challenges. Lean Approach: Waste Management, Kaizen and Kanban, add process and products add value. Roles related to the lifecycle, differences between Agile and traditional plans, differences between Agile plans at different lifecycle phases. Testing plan links between testing, roles and key techniques, principles, understand as a means of assessing the initial status of a project/ How Agile helps to build quality

UNIT II AGILE AND SCRUM PRINCIPLES

Agile Manifesto, Twelve Practices of XP, Scrum Practices, Applying Scrum. Need of scrum, working of scrum, advanced Scrum Applications, Scrum and the Organization, scrum values

UNIT III AGILE PRODUCT MANAGEMENT

Communication, Planning, Estimation Managing the Agile approach Monitoring progress, Targeting and motivating the team, Managing business involvement, Escalating issue. Quality, Risk, Metrics and Measurements, Managing the Agile approach Monitoring progress, Targeting and motivating the team, Managing business involvement and Escalating issue

UNIT IV AGILE REQUIREMENTS AND AGILE TESTING

User Stories, Backlog Management. Agile Architecture: Feature Driven Development. Agile Risk Management: Risk and Quality Assurance, Agile Tools. Agile Testing Techniques, Test-Driven Development, User Acceptance Test

UNIT V SCALING AGILE FOR LARGE PROJECTS

Agile Metrics and Measurements, The Agile approach to project estimating Agile Control: the 7 control parameters. Agile approach to Risk management, The Agile approach to Configuration Management, The Atern Principles, AternPhilosophy, Refactoring, Continuous integration, Automated Build Tools.

`

References

1.Robert C. Martin, Agile Software Development- Principles, Patterns and Practices, Prentice Hall, 2013.
2. Kenneth S. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison Wesley, 2012.
3. James Shore and Shane Warden, The Art of Agile Development, O'Reilly Media, 2007.
4. Craig Larman, ―Agile and Iterative Development: A manager's Guide, Addison-Wesley, 2004.
5. Ken Schawber, Mike Beedle, Agile Software Development with Scrum, Pearson, 2001.
6. Cohn, Mike, Agile Estimating and Planning, Pearson Education, 2006.
7. Cohn, Mike, User Stories Applied: For Agile Software Development Addison Wisley, 2004.

E-references

1. "Agile Software Development", https://www.edx.org/course/agile-software-development

2. "Agile Software Development", https://www.coursera.org/learn/agile-software-development

3. "The Complete Guide to Agile Software Development" https://clearbridgemobile.com/completeguideagile-software-development

4. "Agile Fundamentals Ebook: A Complete Guide for Beginners", https://agileken.com/agilefundamentals-ebook/

Suggested List of Experiments

- Prepare a simple project with its objectives vision statement use case and UML diagrams, banking system, E-Commerce, Healthcare
- Design project release map, user stories for the same
- Design product road map for the proposed project
- Design story mapping for the proposed project
- Compile the product blocking for the proposed project
- Compile the spirit backlog for the proposed project

`

BTech.CSE, V SEM

Machine Learning (CS 502)

COURSE OUTCOMES:

After Completing the course student should be able to

CO1	To understand the fundamental concept of Machine Learning and its differentiation from traditional programming.
CO2	To explore various supervised algorithms including Linear Regression, Logistic Regression, Decision Trees, Random Forests, and k-
CO3	To understand the need and application of unsupervised learning for pattern discovery and data grouping.
CO4	To explore the fundamentals of Reinforcement Learning including agent, environment, and reward systems.
CO5	To study the ML model lifecycle including training, testing, validation, deployment, and monitoring.

Course contents

Unit 1: Introduction to Machine Learning

What is Machine Learning: Definition and scope. Types of Machine Learning: Supervised, Unsupervised, Semi-Supervised, Reinforcement Learning. Applications of ML: NLP, CV, Healthcare, Finance, Robotics, etc. ML vs Traditional Programming. ML Lifecycle. Libraries for ML: NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow (Intro).

Unit 2: Supervised Learning Algorithms

Regression Algorithms: Linear Regression (Simple & Multiple), Polynomial Regression, Regularization Techniques: Lasso, Ridge, ElasticNet. Classification Algorithms: Logistic Regression, k-Nearest Neighbors (k-NN), Decision Trees and Random Forests, Support Vector Machines (SVM). Model Evaluation Techniques, Confusion Matrix, Accuracy, Precision, Recall, F1-Score

Unit 3: Unsupervised Learning and Dimensionality Reduction

Clustering – k-Means, Hierarchical, DBSCAN, Association Rule Mining – Apriori, FP-Growth, Dimensionality Reduction – Feature Selection, Feature Extraction, Principal Component Analysis (PCA), t-SNE (intro), Anomaly Detection – Statistical and ML-based techniques

Unit 4: Ensemble Methods and Model Optimization

Ensemble Learning – Bagging, Boosting, Random Forests, AdaBoost, Gradient Boostin (overview), Model Selection – Grid Search, Random Search, Hyperparameter Tuning, Underfitting, Overfitting, Cross-Validation.

Unit 5: Neural Networks and Advanced Topics

Perceptron Model, Multilayer Perceptron (MLP), Activation Functions –Softmax, Backpropagation, Gradient Descent. Introduction to Deep Learning – CNNs, RNNs (basic), Transfer Learning. Reinforcement Learning – MDPs, Q-Learning, Ethics in ML – Fairness, Accountability, Transparency, Explainable AI.

`

References

1. Christopher M. Bishop, “Pattern Recognition and Machine Learning”, Springer-Verlag New York Inc., 2nd Edition, 2011.
2. Tom M. Mitchell, “Machine Learning”, McGraw Hill Education, First edition, 2017.
3. Ian Goodfellow and YoshuaBengio and Aaron Courville, “Deep Learning”, MIT Press, 2016
4. Aurelien Geon, “Hands-On Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems”, Shroff/O'Reilly; First edition (2017)

Suggested List of Experiment

1. Write a program to practice Python fundamentals, statements, no typed variables, control statements
2. Write a program to practice Python function declaration, definition and call
3. Write a program to hands on with dictionary, list, tuples of python
4. Write a program to create and manipulate NumPy arrays for basic numerical operations.
5. Write a program to load and analyze a dataset using Pandas (e.g., find mean, median, mode, and correlations).
6. Write a program to visualize data using Matplotlib and Seaborn (scatter plot, histogram, heatmap).
7. Write a program to implement simple linear regression using Scikit-learn and visualize the regression line.
8. Write a program to implement multiple linear regression and evaluate it using R^2 and MSE.
9. Write a program to implement polynomial regression and compare with linear regression.
10. Write a program to implement Lasso and Ridge regression for regularization and interpret the results.
11. Write a program to perform logistic regression and predict binary outcomes (e.g., spam vs ham).
12. Write a program to implement k-Nearest Neighbors (k-NN) classification on a dataset.
13. Write a program to evaluate classification models using confusion matrix, precision, recall, and F1-score.
14. Write a program to implement k-Means clustering and visualize the clusters.
15. Write a program to demonstrate model selection using Grid Search and Cross-Validation.

Additional

Write a program to build and visualize a Decision Tree classifier using Scikit-learn.
Write a program to implement a Random Forest classifier and analyze feature importance.
Write a program to apply Support Vector Machine (SVM) classification with different kernels.
Write a program to perform hierarchical clustering and display the dendrogram.
Write a program to apply association rule mining using the Apriori algorithm.
Write a program to apply Principal Component Analysis (PCA) for dimensionality reduction and visualize results.
Write a program to demonstrate model selection using Grid Search and Cross-Validation.
Write a program to implement a simple neural network using Scikit-learn or Keras for classification.
Write a program to implement Q-learning (conceptual demonstration) using a simple environment.

`

BTech.CSE, V SEM

Salesforce I (CS 503(a))

COURSE OUTCOMES:

After Completing the course student should be able to

CO1	Understand the Fundamentals of Cloud Computing and Salesforce CRM
CO2	Analyse Salesforce Architecture and Data Modelling
CO3	Develop Proficiency in Data Management and Migration
CO4	Implement Robust Security and Access Management
CO5	Develop and Analyse Reports and Dashboards and Apex codes

Course Contents

UNIT – I

Overview of Cloud Computing, Introduction to Salesforce, Benefits of Salesforce in CRM,
Salesforce Architecture Overview, Understanding Objects, Fields, and Relationships, Data Model Design Best Practices

UNIT – II

Types of objects, Types of Relationships, (Master-Detail, Look-up, self, etc.), Types of Fields, DataImport and Export Tools in Salesforce, Data Quality Management, Data Migration Strategies andBest Practices

UNIT – III

User Management (Profiles, Roles, Permission Sets), Organization-Wide Defaults and Sharing
Rules, Role Hierarchies and Record Access, Field-Level Security, Salesforce Security Best Practices,

UNIT – IV

Workflow Rules and Process Builder, Approval Processes, Validation Rules, Lightning Flow (FlowBuilder), Creating Workflow Rules and Approval Processes, Designing Automation with LightningFlow, Creating and Customizing Reports, Report Types and Formats, Dashboards: Components andBest Practices, Scheduling and Subscribing to Reports

UNIT – V

Apex basics: development setup, syntax, data types, control structures, OOP concepts, exception handling, SOQL/DML operations, governor limits, and debugging—emphasizing best practices for writing efficient, scalable Salesforce code.

Reference Books:

1. Salesforce For Dummies" by Tom Wong, Liz Kao, and Matt Kaufman

2. "Salesforce CRM – The Definitive Admin Handbook" by Paul Goodey

3. "Mastering Salesforce CRM Administration" by Rakesh Gupta

4. "Salesforce Lightning Platform Enterprise Architecture" by Andrew Fawcett

`

Suggested List of experiments

1.Create a Salesforce Developer Account and explore Salesforce CRM features.

2. Create custom objects, custom fields and design object relationships (Master-Detail,Lookup, Self).

3. Use Schema Builder to visualize object relationships and data model design.

4. Use Data Import Wizard to import sample data into Salesforce custom objects.

5. Use Data Loader for bulk data import and export operations.

6. Create Validation Rules, Duplicate Rules and Matching Rules for data quality management.

7. Create and assign Profiles, Roles, Permission Sets, and Role Hierarchies for usermanagement and record-level security.

8. Configure Organization-Wide Defaults (OWD), Sharing Rules, and Manual Sharing to controlrecord access.

9. Create Workflow Rules and Process Builder Flows to automate business processes with fieldupdates and email alerts.

10. Create multi-step Approval Processes with email notifications and approval hierarchies.

11. Design advanced business automation using Lightning Flow (Flow Builder): Screen Flows,Record-Triggered Flows.

12. Create and customize various types of Reports (Tabular, Summary, Matrix) and Dashboardsusing Report Builder.

13. Schedule Reports and Dashboard Subscriptions for automated reporting.

14. Write simple Apex classes and triggers to update records automatically on insert/update.

15. Write SOQL queries, perform DML operations, handle exceptions, and analyze governorlimits using Developer Console.

`

B.Tech.CSE, V SEM

Cloud computing (CS-503-b)

COURSE OUTCOMES:

After Completing the course student should be able to

CO1	Explain the core concepts of the cloud computing paradigm
CO2	Apply fundamental concepts in cloud infrastructures
CO3	Discuss system, network and storage virtualization
CO4	Illustrate the fundamental concepts of cloud storage
CO5	Analyze various cloud program and setups

Course Contents

UNIT-1.

Introduction of Cloud Computing: What is Cloud Computing?, How it works?, Types of Cloud, Goals & Challenges, Leveraging Cloud Computing, Cloud Economics and Total Cost of Ownership, Cloud Service Models Software as a Service (SaaS): Introduction, Challenges in SaaS Model, SaaS Integration Services, Advantages and Disadvantages. Infrastructure As a Services (IaaS): Introduction, Virtual Machines, VM Migration Services, Advantages and Disadvantages. Platform As a service (PaaS): Introduction, Integration of Private and Public Cloud, Advantages and Disadvantages

UNIT-II

Virtualization and Abstraction: What is Virtualization and how abstraction is provided in cloud? Advantages and Disadvantages, Types of Hypervisor, and Load balancing.

UNIT-III

Amazon Web Services Getting started with AWS, AWS Compute, Storage, and Networking, AWS Security, Identity, and Access Management, AWS Database Options, AWS Elasticity and Management Tools

UNIT-IV

Architecting on AWS Introduction to System Design: AWS Essentials Review and System Design for High Availability, Automation and Serverless Architectures: Event-Driven Scaling, Well-Architected Best Practices: Security, Reliability, Performance Efficiency, Cost Optimization and Deployment and Implementation: Design Patterns and Sample Architectures

UNIT-V

Cloud Security Tools and technologies to secure the data in Private and Public Cloud Architecture. Security Concerns, Legal issues and Aspects, Multi-tenancy issues, Cloud Simulation

`

REFERENCES:

1. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010
2. Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wile, 2011Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012
3. Cloud Security: A Comprehensive Guide to Secure Cloud Computing, s, Wiley-India, 2010

Suggested List of experiments

- Implementation of storage as a Service on cloud through GoogleDrive. working with google docs and google spreadsheets CO1
- Installation and hands on of Google App Engine CO2
- Study and Implementation of virtualization through any Hypervisor. CO3
- Prepare any one Case Study on Cloud Computing Architecture Design for i) Banking System ii)Smart Agriculture iii) Ronald L. Krutz, Russell Dean VineSmart Education CO4
- Installation and Configuration of Hadoop/Eucalyptus CO5
- Hands on AWS, Microsoft Azure CO5

`

B.Tech.CSE, V SEM

Cyber Security (CSE/AIML- 504)

COURSE OUTCOMES:

After Completing the course student should be able to

CO1	Students will be able to understand the concept of cybercrime
CO2	Students will be able to analyze different types of cyber offenses
CO3	Students will be able to examine the legal provisions related to cybercrimes under the IT Act, 2000
CO4	Students will be able to compare the Indian Evidence Act, 1872 with the IT Act, 2000, and assess the legal status of electronic
CO5	Students will be able to identify and explain common cybercrime tools and methods

Course Contents

UNIT 1: Introduction to Cyber Crime :: Introduction of Cyber Crime, Challenges of cyber crime, Classifications of Cybercrimes: EMail Spoofing, Spamming, Internet Time Theft, Salami attack/Salami Technique,

UNIT 2: Categories and Perception of Cyber Crimes :: Web jacking, Online Frauds, Software Piracy, Computer Network Intrusions, Password Sniffing, Identity Theft, cyber terrorism, Virtual Crime, Perception of cyber criminals: hackers, insurgents and extremist group etc. session hijacking.

UNIT 3 : Cyber Crime and Legal Framework :: Cyber Crime and Criminal justice: Concept of Cyber Crime and the IT Act, 2000, Hacking, Teenage Web Vandals, Cyber Fraud and Cheating, Defamation, Harassment and E-mail Abuse, Other IT Act Offences, Monetary Penalties, jurisdiction and Cyber Crimes, Nature of Criminality, Strategies to tackle Cyber Crime and Trends.

UNIT 4 : Cyber Evidence and Legal Admissibility :: The Indian Evidence Act of 1872 v. Information Technology Act, 2000: Status of Electronic Records as Evidence, Proof and Management of Electronic Records; Relevancy, Admissibility and Probative Value of E-Evidence, Proving Digital Signatures, Proof of Electronic Agreements, Proving Electronic Messages.

UNIT 5 : Cybercrime Tools and Techniques :: Tools and Methods in Cybercrime: Proxy Servers and Anonymizers, Password Cracking, Key loggers and Spyware, virus and worms, Trojan Horses, Backdoors, DoS and DDoS Attacks , Buffer and Overflow, Attack on Wireless Networks, Phishing : Method of Phishing, Phishing Techniques.

REFERENCES:

1. Principles of Cyber crime, Jonathan Clough Cambridge University Press
2. John R. Vacca, Computer Forensics:Computer Crime Scene Investigation, 2nd Edition, Charles River Media, 2005
3. Cyber Law Simplified, VivekSood, Pub: TMH.
4. Cyber Security by Nina Godbole, SunitBelapure Pub: Wiley-India
5. Information Warfare: Corporate attack and defense in digital world, William Hutchinson, Mathew Warren, Elsevier.
6. Cyber Laws and IT Protection, Harish Chander, Pub:PHI.

`

CS/AL 506 Minor Project Syllabus

The minor project focuses on introducing students to the fundamentals of full stack development. Students will begin by understanding the architecture of web applications, including how the frontend and backend communicate. They will then design a simple user interface that captures and displays data dynamically. The backend will include basic server setup, data handling, and implementation of CRUD (Create, Read, Update, Delete) operations with a connected database. Emphasis will be placed on developing clean code, maintaining basic security practices, and validating user input. Finally, students will test the application for core functionalities, deploy it to a hosting platform, and present a concise documentation highlighting their learning journey.

Major Project Syllabus

The major project involves designing and developing a comprehensive full stack application that solves a real-world problem. Students will start with requirement analysis, system design, and data modeling for the application. The frontend development will include dynamic pages, routing, user authentication, and effective state management. The backend will handle complex logic, secure data transactions, and seamless integration with the database using APIs. The project will include robust testing, performance tuning, and deployment on a scalable hosting environment. A detailed project report, including system architecture, challenges faced, and future enhancements, will be submitted along with a live demonstration.