303105300 - Project - 1

Course: BTech **Semester:** 6

Rationale : : One of the important criteria of "Project " is to develop the ability of "learning to Learn " on its own. This would go a long way helping the students in keeping pace with future changes in technology and in the acquisition of knowledge and skills as and when needed. The course of the "Project" is designed with an aim to all these requirements of the students. Which will include planning of the Programme, which must be completed within the time allocated. The Project should never have a single solution and process of arriving at a particular solution, the student must be required to make number of decisions after study information as he has gathered from experiments, surveys, analysis etc.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	6	-	3	-	-	50	-	50	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introductory Guideline:**General instruction about project definition, different platform etc.	10	15
2	**Analysis:**Deep study about project title and available system and user requiremen	20	15
3	**Design:**Designing include all requirement gathered in analysis part	20	15
4	**Implementation:**Implement your design work	20	15
5	**Testing:**Different test case must implement for your project	15	15
6	**Documentation:**Project report	15	15
	Total	100	90

Course Outcome

After Learning the Course the students shall be able to:

1. Define characteristics of project.
2. Manage project plan, monitor and controlling project schedule and budget, tracking project progress 3. Deliver a seminar on the general area of work being undertaken and specific contributions to that field.
4. Prepare a formal report describing the work undertaken and results obtained so far.
5. Present the work in a forum involving poster presentations and demonstrations of operational hardware and software.

303105349 - Compiler Design

Course: BTech **Semester:** 6

Prerequisite: Algorithms, Data Structures, Assembly Language Program, Theory of Computation, C/C++ Programming Skills | 203105205 - Data Structure and Algorithms

Rationale : Compiler Design is a fundamental subject of Computer Engineering. Compiler design principles provide an in-depth view of translation, optimization and compilation of the entire source program. It also focuses on various designs of compiler and structuring of various phases of compiler. It is inevitable to grasp the knowledge of various types of grammar, lexical analysis, yacc, FSM(Finite State Machines) and correlative concepts of languages.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Overview of compilation :** The structure of a compiler and applications of compiler technology; Lexical analysis - The role of a lexical analyzer, specification of tokens, recognition of tokens, hand-written lexical analyzers, LEX, examples of LEX programs.	10	8
2	**Introduction to syntax analysis** Role of a parser, use of context-free grammars (CFG) in the specification of the syntax of programming languages, techniques for writing grammars for programming languages (removal left recursion, etc.), non-context-free constructs in programming languages, parse trees and ambiguity, examples of programming language grammars.	10	7
3	**Top-down parsing** FIRST & FOLLOW sets, LL(1) conditions, predictive parsing, recursive descent parsing, error recovery. LR-parsing - Handle pruning, shift-reduce parsing, viable prefixes, valid items, LR(0) automaton, LR-parsing algorithm, SLR(1), LR(1), and LALR(1) parsing. YACC, error recovery with YACC and examples of YACC specifications.	20	7

4	**Syntax-directed definitions (attribute grammars)** Synthesized and inherited attributes, examples of SDDs, evaluation orders for attributes of an SDD, dependency graphs. S-ttributed and L-attributed SDDs and their implementation using LR-parsers and recursive-descent parsers respectively.	**15**	**6**
5	**Semantic analysis** Symbol tables and their data structures. Representation of "scope". Semantic analysis of expressions, assignment, and control-flow statements, declarations of variables and functions, function calls, etc., using S- and L-attributed SDDs (treatment of arrays and structures included). Semantic error recovery.	**15**	**6**
6	**Intermediate code generation** Different intermediate representations –quadruples, triples, trees, flow graphs, SSA forms, and their uses. Translation of expressions (including array references with subscripts) and assignment statements. Translation of control-flow statements – it- then-else, while-do, and switch. Short-circuit code and control-flow translation of Boolean expressions. Back patching. Examples to illustrate intermediate code generation for all constructs.	**15**	**6**
7	**Run-time environments**Stack allocation of space and activation records. Access to non-local data on the stack in the case of procedures with and without nesting of procedures.	**10**	**3**
8	**Introduction to machine code generation and optimization** Simple machine code generation, examples of machine-independent code optimizations.	**5**	**2**
	Total	**100**	**45**

Reference Books	
1.	**Compilers: Principles, Techniques and Tools** By Aho, Lam, Sethi, and Ullman | Pearson | Second, Pub. Year 2014

Course Outcome
After Learning the Course the students shall be able to:
1. Understand the basic concepts; ability to apply automata theory and knowledge on formal languages. 2. Ability to identify and select suitable parsing strategies for a compiler for various cases. Knowledge in alternative methods (top- down or bottom-up, etc.). 3. Understand backend of compiler: intermediate code, Code optimization Techniques and Error Recovery mechanisms 4. Understand issues of run time environments and scheduling for instruction level parallelism.

303105350 - Compiler Design Laboratory

Course: BTech **Semester:** 6

Prerequisite: Algorithms, Data Structures, Assembly Language Program, Theory of Computation, C/C++ Programming Skills | 203105205 - Data Structure and Algorithms

Rationale : Compiler Design is a fundamental subject of Computer Engineering. Compiler design principles provide an in-depth view of translation, optimization and compilation of the entire source program. It also focuses on various designs of compiler and structuring of various phases of compiler. It is inevitable to grasp the knowledge of various types of grammar, lexical analysis, yacc, FSM (Finite State Machines) and correlative concepts of languages.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
After learning the course, the students should be able to 1. Understand the basic concepts; ability to apply automata theory and knowledge on formal languages. 2. Ability to identify and select suitable parsing strategies for a compiler for various cases. Knowledge in alternative methods (top-down or bottom-up, etc.). 3. Understand backend of the compiler: intermediate code, Code Optimization Techniques and Error Recovery mechanisms 4. Understand issues of run time environments and scheduling for instruction level parallelism

List of Practical

1.	**Program to implement Lexical Analyzer.**
2.	**Program to count digits, vowels and symbols in C.**
3.	**Program to check validation of User Name and Password in C.**
4.	**Program to implement Predictive Parsing LL (1) in C.**
5.	**Program to implement Recursive Descent Parsing in C.**
6.	**Program to implement Operator Precedence Parsing in C.**
7.	**Program to implement LALR Parsing in C.**
8.	**To Study about Lexical Analyzer Generator (LEX) and Flex (Fast LexicalAnalyzer)**

9.	**Implement following programs using Lex.** a. Create a Lexer to take input from text file and count no of characters, no. of lines & no. of words. b. Write a Lex program to count number of vowels and consonants in a given input string.
10.	**Implement following programs using Lex.** a. Write a Lex program to print out all numbers from the given file. b. Write a Lex program to printout all HTML tags in file. c. Write a Lex program which adds line numbers to the given file and display the same onto the standard output.

303105379 - Mobile App Development

Course: Btech **Semester:** 6

Prerequisite: Basic knowledge of java language

Rationale : The mobile application development syllabus covers the essential concepts and tools for building apps across platforms, including UI/UX design, app architecture, networking, databases, and deployment. It explores both native development (Android) and cross-platform frameworks, emphasizing practical skills for creating functional, user-friendly mobile applications.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Android Operating System and Development Environment :** Introduction, Android Architecture, Versions, Features, OHA, Dalvik VM, Android SDK, Android Development Tools, Android Virtual Devices, Development Environment, Directory Structure of Android Application, Android Manifest file	10	3
2	**Android Components and Resource handling :** Components: Context, Activity, Intent, Service, Broadcast Receiver, Resources:String, Color, Drawable, Styles, Theme, Localization:Prepare Application for Localization	20	7
3	**Android User Interface Elements and Layouts:** Introduction of Material Design, UI and UX **Layouts:** Linear Layout, Absolute Layout, Frame Layout, Relative Layout, Constraint Layout, Dynamic Implementation of Layout. UI widgets with properties, events and methods, Dialog boxes, Menus: Option and Context	20	8
4	**Working with Views and Fragment:** GridView, WebView, ScrollView, ListView, RecyclerView, CardView Fragment: Introduction, life Cycle, Implementation	10	5
5	**Data Storage Techniques :** Shared Preferences, Files and Directories, SQLite Database Connectivity andOperations, Content Providers: Basics, Content URI, Content Resolver, Built-in content providers.	20	9

6	**Web Application Integration Techniques:** Introduction of AsyncTask, Communication with Web API, Introduction to JSON data, JSON Parsing,Implementation of Third-PartyLibraryto Fetch Network Data, Notifications, Telephony API, Google API	**10**	**8**
7	**Polish and Publish Application:** Different Ways to Monetize, Versioning, Signing, Packaging and Beta Test of Mobile Application, Distributing Application on Mobile Market Place	**10**	**5**
	Total	**100**	**45**

Reference Books

1.	**Android Wireless Application Development By Lauren Darcey and Shane Conder | Pearson Education,2011 | second edition (TextBook)**
2.	**Head First Android Development: A Brain Friendly Guide, O`Reilly, David Griffiths and Dawn Griffiths**
3.	**Professional Android 4 Application Development, John Wiley & Sons Author(s): Reto Meier**
4.	**Beginning Android, Apress Author(s): Mark L Murphy**

Course Outcome

After Learning the Course the students shall be able to:
1. Acquire an insight into concepts of mobile application development terminologies, environment and architecture 2. Design mobile application using various UI components and layouts. 3. Develop robust mobile applications with database interaction and webservice integration 4. Deploy application on mobile device

303105379 - Mobile App Development Laboratory

Course: Btech **Semester:** 6

Prerequisite: Basic knowledge of java language

Rationale : The mobile application development syllabus covers the essential concepts and tools for building apps across platforms, including UI/UX design, app architecture, networking, databases, and deployment. It explores both native development (Android) and cross-platform frameworks, emphasizing practical skills for creating functional, user-friendly mobile applications.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Acquire an insight into concepts of mobile application development terminologies, environment and architecture 2. Design mobile application using various UI components and layouts. 3. Develop robust mobile applications with database interaction and webservice integration 4. Deploy application on mobile device.

List of Practical

1.	**Create a "Hello World" application: Display "Hello World" at the center of the screen, both on the Android emulator and an actual Android device.**
2.	**Build an app to showcase Android lifecycle phases: Develop an app that demonstrates various Android lifecycle stages (onCreate, onStart, onResume, etc.).**
3.	**Create an app with two activities: The first activity should contain an EditText and a "Send" button. When the button is clicked, use an explicit intent to send the text from EditText to a second activity and display it in a TextView.**
4.	**Create an app with explicit intent: The first activity should have an EditText and a "Send" button. On button click, use an implicit intent with the "SEND" action, allowing the user to select an app from an app chooser to handle the intent and display the text.**
5.	**Build a basic calculator app: Create an app that performs basic arithmetic operations (addition, subtraction, multiplication, and division) on numbers.**

6.	Create a Spinner-based app: Develop an app with a spinner populated from the res/values/strings.xml resource. When the spinner value changes, the corresponding image from the res/drawable directory should be displayed.
7.	Create a discount calculator app: Use a RadioGroup with three radio buttons for 10%, 15%, and 20% discounts on a shopping bill. The user can enter the bill amount in an EditText, and the selected discount will be calculated and displayed in a TextView.
8.	Create an app with a course selection RadioButton group: Display a list of college courses with a RadioButtongroup. When a course is selected, the corresponding TIC (Total Instructional Credit) should be shown in a TextView.
9.	Create a shopping list app using checkboxes: Build an app with checkboxes for shopping list items. As items are checked off, the selected items should be displayed in a TextView.
10.	Create a login and registration app: Develop a login application that verifies the username and password. Include a registration page for new users. Upon successful login, show a "Welcome User" pop-up message
11.	Create a login app with navigation to another activity: The login screen should verify the username and password. After successful login, navigate to a new activity that displays a "Welcome User" message in a TextView and a "Logout" button. On clicking "Logout," show a confirmation dialog with "OK" and "Cancel" buttons. "OK" should return to the login screen, while "Cancel" should keep the user on the current activity.
12.	Create an app with a menu: Implement a menu with five options. The selected option should be displayed in a TextView.
13.	Build an app using LinearLayout: Create a simple app that uses LinearLayout. It should take the contents of a predefined TextView, convert it to uppercase on button click, and display it in an EditText. Additionally, create an app that responds to key events in the EditText without needing a button press.
14.	Create an app with TableLayout and custom styles: Use a TableLayout with a TextView, EditText, and buttons. Also, create a custom styles.xml in the res/values directory to style the TextView.
15.	Create an app with SQLite database operations: Build an app that allows the user to perform CRUD operations (Create, Read, Update, Delete) with an SQLite database.
16.	Create an app with three vertically aligned buttons: Develop an app with three buttons arranged vertically. When any button is selected, the screen color should change accordingly.

Course: Btech **Semester:** 6

Prerequisite: Database Management system, SQL, Basics of Javascript and web development

Rationale : 1. Understanding the basics of web development and JavaScript programming 2. Learning how to use MongoDB, a popular NoSQL database, to store and retrieve data 3. Learning how to use Node.js, a server-side JavaScript runtime, to create APIs and handle server-side logic 4. Learning how to use Express.js, a lightweight web application framework for Node.js, to build web applications 5. Learning how to use AngularJS, a powerful front-end JavaScript framework, to create dynamic user interfaces and connect with APIs 6. Building a full-stack web application from scratch using the MEAN stack 7. Understanding best practices for deploying, testing, and maintaining MEAN stack applications

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	0	0	-	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W
1	**Introduction to Web Development and the MEAN Stack:** Overview of web development, Introduction to the MEAN stack, Setting up the development environment	4
2	**MongoDB:** Introduction to NoSQL databases, Installation and configuration of MongoDB, CRUD operations in MongoDB, Indexing and querying in MongoDB, Schema design and data modeling	20
3	**Node.JS & Express JS:** Introduction to Node.js and Express.js, Introduction to Node.js and Express.js, Middleware and routing, Authentication and security with Passport.js, Error handling and logging	20
4	**Angular:** Introduction to Angular, Setting up an Angular application, Components, modules, and services, Data binding and templates, Forms and validation, Routing and navigation, HTTP and observables, Building a complete frontend for the MEAN stack application	30
5	**Integration:** Integrating the Angular frontend with the Express.js API, Authentication and user management integration , Handling real-time data with WebSockets, Error handling and testing	10
6	**Deployment and Best Practices:** Preparing the application for deployment, Hosting and server setup options, Security best practices, Performance optimization and testing, Version control and continuous integration.	

Reference Books	
1.	**MEAN Web Development" by Amos Q. Haviv (Publisher: Packt Publishing) (TextBook)**
2.	**"Learning Node.js: A Hands-On Guide to Building Web Applications in JavaScript" by Marc Wandschneider (Publisher: Addison-Wesley Professional)**
3.	**"AngularJS: Up and Running: Enhanced Productivity with Structured Web Apps" by Shyam Seshadri and Brad Green (Publisher: O'Reilly Media)**
4.	**"MongoDB: The Definitive Guide: Powerful and Scalable Data Storage" by Shannon Bradshaw, Kristina Chodorow, and Eoin Brazil (Publisher: O'Reilly Media)**
7	**Final Project:** Project

Course Outcome
After Learning the Course the students shall be able to:
1. Have a comprehensive understanding of the technologies and frameworks that make up the MEAN stack, including MongoDB, Expíess.js, AngularJS, and Node.js. 2. Build full-stack web applications. 3. Understand web development best practices: 4. Work on real-world projects using the MEAN stack. This could include developing a portfolio of projects of contributing to open- source projects.

303105386-MEA(R)N Stack Web Development Laboratory

Course: BTech **Semester:** 6

Prerequisite: Database Management system, SQL, Basics of Javascript and web development

Rationale : 1. Understanding the basics of web development and JavaScript programming 2. Learning how to use MongoDB, a popular NoSQL database, to store and retrieve data 3. Learning how to use Node.js, a server-side JavaScript runtime, to create APIs and handle server-side logic 4. Learning how to use Express.js, a lightweight web application framework for Node.js, to build web applications 5. Learning how to use AngularJS, a powerful front-end JavaScript framework, to create dynamic user interfaces and connect with APIs 6. Building a full-stack web application from scratch using the MEAN stack 7. Understanding best practices for deploying, testing, and maintaining MEAN stack applications

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	-	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Have a comprehensive understanding of the technologies and frameworks that make up the MEAN stack, including MongoDB, Expíess.js, AngularJS, and Node.js. 2. Build full-stack web applications. 3. Understand web development best practices: 4. Work on real-world projects using the MEAN stack. This could include developing a portfolio of projects of contributing to opensource projects.

List of Practical

1.	**1. Introduction to MEAN stack 2. Setting up the development environment 3. Overview of MongoDB, Express.js, Angular, and Node.js**
2.	**1. Creating and configuring MongoDB 2. Creating and configuring Express.js 3. Building RESTful APIs with Express.js**
3.	**1. Introduction to Angular 2. Building basic UI components with Angular 3. Creating a Single-Page Application (SPA) with Angular**

4.	**1. Introduction to Node.js 2. Creating and configuring Node.js 3. Building server-side applications with Node.js**
5.	**1. Integrating all components to build a full-stack application 2. Testing and debugging the application 3. Deploying the application on a cloud platform**

303193353 - Employability Skills

Course: BTech **Semester:** 6

Prerequisite:

Rationale : -

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
-	1	-	-	1	-	100	-	-	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**IELTS Mock Test** To develop students English Learning and improve their employment prospects. To create opportunity for students to study around the globe & give them Practice on : Listening Speaking Reading Writing	25	5
2	**Resume Building** Cover letter & Resume Writing Students will create a functional resume along with cover letter that they will be able to use when applying for a job, college or a scholarship.	25	2
3	**Advanced Group Discussion: Mock Round** To provide students with an avenue to train themselves in various interpersonal skills. To prepare students for the Group Discussion after the written test for employment or for admission to educational institutes. To generate new ideas or new approaches for solving a problem. To reach a solution on an issue of concern.	25	4
4	**Personal Interview: Mock Round** Preparing For The Interview Review Question Employer's Expectation Case Interview	25	4
	Total	100	15

Reference Books

1.	**Business Correspondence and Report Writing** By SHARMA, R. AND MOHAN, K.
2.	**Communication Skills and Soft Skills** By Suresh Kumar | Pearson Publication, 2010

Prerequisite: Basic object-oriented programming concepts.

Rationale : The objective of this course is to provide a conceptual understanding of how blockchain technology can be used to innovate and improve business processes. The course covers the technological underpinning of blockchain operations in the both theoretical and practical implementation of solutions using block Chain technology.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Blockchain in Software Architecture:** What Is Blockchain and Why Should I Care?, Defining Blockchain, Smart Contracts and Decentralized Applications, cryptocurrencies and Tokens, Blockchain-Based Applications, Enterprise and Industry, Financial Services, Government Services, Blockchain Functionality, Blockchain as Data Storage, Transactions, Digital Assets, Blockchain as a Computational Infrastructure, Blockchain Non-functional Properties, Non-functional Properties and Requirements, Non- functional Properties of Blockchain, Blockchain Architecture Design, Software Architecture: Design and Analysis, Designing Blockchain-Based Applications	20	9
2	**Existing Blockchain Platforms:** Bitcoin Transactions, Script, Mining, Accounts and State, Nakamoto Consensus, Deflationary Cryptocurrency, Wallets, Exchanges, Ethereum, Ethereum Protocol, Ethereum Transactions, Smart Contract, Decentralized Application (dapp), Hyperledger Fabric, Permissioned Blockchain, Chain code as Smart Contract, Nodes, Transactions, Consensus, Other Representative Blockchain Platforms	20	9
3	**Varieties of Blockchains:** Fundamental Properties of Blockchain, Decentralization, Permission, Deployment, Ledger Structure, Consensus Protocol, Block Configuration, Auxiliary Blockchains, Anonymity, Incentives	20	9
4	**Use Cases:** Key Non-functional Requirements, Conventional Technology, A Blockchain Solution, Non-functional Property Discussion, Open Data Registry, Key Non-functional Requirements, Conventional Technology, A Blockchain Solution, Non-functional Property Discussion,International Money Transfers, Key Non-functional Requirements, Conventional Technologies, Blockchain Solution, Non-functional Property Discussion	20	9
5	**Blockchain in Database:** Blockchain as an Architectural Element, Blockchain as Storage Element, Comparison with Centralized Databases, Comparison with Cloud Services, Comparison with Peer-to-Peer Data Storage, Comparison with Replicated State Machines, Blockchain as Computational Element, Blockchain as Communication Mechanism	20	9
	Total	100	45

Reference Books

1.	**Architecture for blockchain applications / Xiwei Xu, Ingo Weber, Mark Staples. (TextBook)**
2.	**Zharg, Derek, "Trustless Trust with Verifiable Computation" (2018)**
3.	**Mohan, C. "Blockchains and databases: A new era in distributed computing." In 2018 IEEE 34th International Conference on Data Engineering (ICDE), pp. 1739-1740. IEEE, 2018.**

Course Outcome

After Learning the Course the students shall be able to:
1. Define the key characteristics of Blockchains. 2. Explain the different layers of components that compose the architecture of a blockchain-based system. 3. Compare a blockchain-based system with a replicated state machine. 4. Understand and evaluate the components of blockchain-based technologies which support Turing- complete languages

Course: BTech

303105344 -Blockchain Components and Architecture Lab

Prerequisite: Basic object-oriented programming concepts.

Rationale : The objective of this course is to provide a conceptual understanding of how blockchain technology can be used to innovate and improve business processes. The course covers the technological underpinning of blockchain operations in the both theoretical and practical implementation of solutions using block Chain technology.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Define the key characteristics of Blockchains.
2. Explain the different layers of components that compose the architecture of a blockchain-based system.
3. Compare a blockchain-based system with a replicated state machine.
4. Understand and evaluate the components of blockchain-based technologies which support Turing- complete languages

List of Practical

1.	**Understand blockchain technology and prepare a short report on it.**
2.	**Develop blockchain-based solutions and write smart contracts using Hyperledger Fabric and Ethereum frameworks.**
3.	**Build and deploy blockchain applications for on-premises and cloud-based architecture.**
4.	**Integrate ideas from various domains and implement them using blockchain technology from different perspectives.**
5.	**Develop blockchain applications.**
6.	**Understand the security features of blockchain technology and develop applications.**

3105348-Parallel Computing

Course: Btech

Prerequisite: Basic programming proficiency in C, C++, and Java.

Rationale : The primary rationale of parallel computing is to increase the available computation power for faster application processing or task resolution.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction to High-Performance Computing:** Overview of High-Performance Computing and Java Framework, Concurrent Techniques, The Benefits of Parallel Programming, The Basic Layers of Concurrent Techniques, Categories of Computers	10	5
2	**Parallel Computer System Architecture:** Overview of Parallel Computing, Central Processing Unit Design, Instruction Set Architecture and Design, Some General Parallel Terminolog	20	9
3	**Microprocessor Technologies:**Overview of Microprocessors, Silicon Technology, Intel Multi Core Technology	10	9
4	**Parallel Computer memory architecture:**Parallel Computer Memory Architectures, SMPs, MPPs And Parallel Processing	20	9
5	**Parallel Programming Model:** Parallel Programming Models, Shared Memory Model, Threads Models, Message Passing Model, Data Parallel Model, Hybrid Programming Model, Flynn's Programming Model, Embarrassingly Parallel Computation, Pipeline Computations	30	8
6	**Designing Parallel Programs:** Automatic Vs Manual Parallelization, Understand The Problem and The Program, Partitioning, Communications, Synchronization, Data Dependencies, Load Balancing, Granularity Parallel Input and Output, Limits and Costs of Parallel Programming, Performance Analysis and Tuning	10	5
	Total	100	45

Reference Books

1.	**Ananth Grama, Anshul Gupta, George Karypis and Vipin Kumar, "Introduction to Parallel Computing", Pearson, 2nd Edition (TextBook)**
2.	**Vipin Kumar, Introduction to Parallel Computing: Design and Analysis of Algorithms**
3.	**Michael j Quinn, Parallel Computing**

Course Outcome

After Learning the Course the students shall be able to:
1. Understand High-Performance Computing 2. Analyze Parallel Computer System Architecture and Memory Architecture 3. Identify Parallel Programming Models 4. Design Parallel Programs

Course: Btech

Prerequisite: Basic programming proficiency in C, C++, and Java.

Rationale : The primary rationale of parallel computing is to increase the available computation power for faster application processing or task resolution.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Understand High-Performance Computing 2. Analyze Parallel Computer System Architecture and Memory Architecture 3. Identify Parallel Programming Models 4. Design Parallel Programs

List of Practical

1.	**Write OpenMP basic programs such as Vector addition, Dot Product**
2.	**Write OpenMP for Loop work-sharing and sections work-sharing**
3.	**Write OpenMP for Combined parallel loop reduction and Orphaned parallel loop reduction**
4.	**Write OpenMP program for Matrix multiply.**
5.	**Write MPI program for Basics of MPI**
6.	**Write MPI program for Communication between MPI process**
7.	**Write MPI program for Collective operation with 'synchronization'**
8.	**Write MPI program for Collective operation with 'data movement'**
9.	**Write MPI program for Collective operation with 'collective computation'**

Course: BTech

303105357 - Programming for IoT

Prerequisite: Basics of electronics circuit and python programming language

Rationale : The course should enable the students to understand the architecture of Internet of Things and connected world. Secondly, it will help the students to explore the use of various hardware and sensing technologies to build IoT applications.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction to IoT:** What Is IoT, Genesis of IoT, IoT and Digitization, Evolutionary Phases of the Internet, IoT Impact, IoT Applications and examples: Connected Roadways, Connected Factory, Smart Connected Buildings, Smart Creatures, Convergence of IT and OT, IoT Challenges.	10	5
2	**The Arduino Environment:** Environment:Introduction to the Arduino environment, the Arduino board, the Arduino IDE, and the Arduinocompatible shields together with their libraries. Arduino board main components, inputs, and outputs. Arduino Integrated Development Environment (IDE), Compiling Code, Arduino Shields and Libraries.	20	9
3	**Arduino Programming:** Composition of Arduino programs, Arduino tool chain, Arduino IDE, basic structure of a sketch, including the use of the setup() and loop() functions. Accessing the pins from a sketch for input and output, introduction on debugging embedded software on an Arduino,	20	9
4	**Arduino Communication Protocols:** UART communication protocol, Synchronization, parity and stop, the use of the Serial library to communicate with the Arduino through the serial monitor, SPI, I2C.	15	8
5	**Advances and Recent trends in IoT through research perspective:**Advances and Recent trends in IoT through research perspective	20	8
6	**Real world problems and their solutions using Arduino:**Real world problems and their solutions using Arduino	15	6
	Total	100	45

Reference Books

1.	**Massimo Banzi, "Getting Started with Arduino", First Edition, February 2009, O'Reilly Media, Inc (TextBook)**
2.	**Michael Margolis, "Arduino Cookbook", First Edition, March 2011, O'Reilly Media, Inc**

Course Outcome
After Learning the Course the students shall be able to:
1. Understand the concept of Internet of Things and connected world. 2. Explore on use of various hardware and sensing technologies to build IoT applications 3. Illustrate the architecture of the Internet of Things. 4. Understand the working of Arduino 5. Explore Interacting with digital outputs.

Course: BTech

303105358 - Programming for IoT Laboratory

Prerequisite: Basics of electronics circuits and python programming

Rationale : The course should enable the students to understand the architecture of Internet of Things and connected world. Secondly, it will help the students to explore the use of various hardware and sensing technologies to build IoT applications.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Understand the concept of Internet of Things and connected world. 2. Explore on use of various hardware and sensing technologies to build IoT applications 3. Illustrate the architecture of the Internet of Things. 4. Understand the working of Arduino 5. Explore Interacting with digital outputs.

List of Practical

1.	**Understanding Arduino hardware and its various pins and protocol support.**
2.	**Arduino Based Alarm Clock**
3.	**Ardunio Based smart traffic light**
4.	**Arduino Based Home Automation**
5.	**Arduino Based Smart Parking System.**
6.	**Arduino Based Health Monitoring System.**
7.	**Arduino Based Home Security System.**
8.	**Arduino Based women safety.**
9.	**Arduino Based Smart City.**
10.	**Attaching analog and digital sensors in Arduino and fetching its values.**
11.	**Using Arduino to push data over cloud, generated by various sensors.**
12.	**Cleaning and analyzing the data.**

Course: BTech

303105359 – Machine Learning for IoT

Prerequisite: Data structure, automata, and languages, Mathematics, Python

Rationale : This course focuses on IoT and how it can be implanted in various real-world scenarios. IoT is all about connecting all the electronic devices to the internet, so as they are connected through the internet we can also control these devices from any place. For controlling electronic devices we need sensors to sense the environment and based on those sensed values these devices are controlled. The sensing of various parameters depends on scenarios. Then these parameters values are analyzed using machine learning and based on those values we can take actions e.g. to start and stop the sprinklers. This is indeed one example but you can apply it to all real-world problems and can give smarter solutions. Thus the aim of this course is to understand a practical problem and then provide smart solutions to it.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Foundations of IoT:** Basics of IoT, Iot Network Architecture and Design, Sensors and Actuators, Connecting Smart Objects	15	6
2	**Exploratory data analysis for IoT domains:** Collecting IoT data using various technologies,Edge/fog/cloud computing for IoT data, Cleaning data, using application layer protocol to fetch the IoT data:MQTT, CoAP.	20	9
3	**Amalgamation of IoT and ML in the Healthcare domain:** Introduction, Background, Various research methodologies used in different aspects of healthcare domain, Various models used in different domains of healthcare, Discussion on various results achieved in the healthcare domain, conclusion	15	6
4	**Application of Machine Learning and IoT for Smart Cities:** Introduction, Background, Various research methodologies used in different aspects of smart city domain, Various models used in different domains of smart city, Discussion on various results achieved in smart city domain, conclusion	15	7
5	**Predicting Air and water quality using IoT and Machine Learning:** Introduction, Background, Various research methodologies used in different parameters of air and water quality, Various models used in the prediction of air and water quality aspect, Discussion on various results achieved, conclusion	15	7
6	**Classification of Attacks on IoT scenarios using ML:** Introduction, Background, Various research methodologies used for intrusion detection and prevention in IoT scenarios, Various models used for intrusion detection and prevention in IoT scenarios using ML, Discussion on various results achieved for intrusion detection and prevention in IoT scenario using ML, conclusion	20	10

Total	**100**	**45**

Reference Books

1.	**A Practical Guidebook to Learn and Implement IoT using Machine Learning (TextBook)** By Purnendu Shekhar Pandey | Perception Publications
2.	**Shalli Rani, R. Maheswar, G. R. Kanagachidambaresan, Sachin Ahuja, Deepali Gupta. Machine Learning Paradigm for Internet of Things Applications, Wiley Publications, 2022.**

Course Outcome

After Learning the Course the students shall be able to:
1. Understand the meaning of IoT and how that can be used in real world scenarios. 2. To fetch the IoT data and clean the data. 3. Decide a suitable model for a given IoT Scenario. 4. Converting prototype models into working models for social cause.

Course: BTech

303105360 - ML for IoT Laboratory

Prerequisite: Data structure, automata, and languages, Mathematics, Python

Rationale : This course focuses on IoT and how it can be implanted in various real-world scenarios. IoT is all about connecting all the electronic devices to the internet, so as they are connected through the internet we can also control these devices from any place. For controlling electronic devices we need sensors to sense the environment and based on those sensed values these devices are controlled. The sensing of various parameters depends on scenarios. Then these parameters values are analyzed using machine learning and based on those values we can take actions e.g. to start and stop the sprinklers. This is indeed one example but you can apply it to all real-world problems and can give smarter solutions. Thus the aim of this course is to understand a practical problem and then provide smart solutions to it.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Understand the meaning of IoT and how that can be used in real world scenarios. 2. To fetch the IoT data and clean the data. 3. Decide a suitable model for a given IoT Scenario. 4. Converting prototype models into working models for social cause.

List of Practical

1.	**Home Automation with Power Management perspective.**
2.	**Predicting Air quality contents using IoT and Machine Learning.**
3.	**Predictive Analysis of Water Quality using IoT and Machine Learning.**
4.	**Smart Farming Using IoT.**
5.	**Multi Object Detection and Color Identification.**
6.	**Speed Alert System at Blind turns and Speed Breaker Detection.**
7.	**Emotion Detection with Image Processing and Pulse Rate Sensor and GSR Sensor.**
8.	**Remotely Monitoring and Controlling Muffle Furnace Using IoT and Image Processing.**
9.	**Remotely Monitoring and Controlling Tensile Testing Machine Using IoT and Image Processing.**
10.	**Hand Gesture Mouse Movement.**
11.	**Deployment of External CPU Cooling Hardware by applying IoT data and Machine Learning Analysis.**

12.	**Classification of Attacks on Raspberry Pi.**
13.	**Overheat Notification of Mobile Phone Battery Using IoT and Machine Learning Analysis.**

Course: BTech

303105364 - Cloud Computing Laboratory

Prerequisite: Fundamentals of Distributed Computing

Rationale : This course aims students to understand the hardware, software concepts and architecture of cloud computing. Students realize the importance of Cloud Virtualization, Abstractions and Enabling Technologies.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	-	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Compare the strengths and limitations of cloud computing 2. Identify the architecture, infrastructure and delivery models of cloud computing 3. Apply suitable virtualization concept. 4. Choose the appropriate cloud player, Programming models and approach 5. Address the core issues of cloud computing such as security, privacy and interoperability

List of Practical

1.	**Understanding single core and multi core Architecture.**
2.	**Understanding Computer Network fundamentals and Designing LANs.**
3.	**Implementation of Infrastructure as a service (IaaS) using Hypervisors.**
4.	**Implementation of private cloud platform using open stack cloud.**
5.	**Working with IaaS of Public cloud platforms.**
6.	**Implementation of Platform as a service (PaaS) in private cloud environment.**
7.	**Implementation Platform as a service (PaaS) in public cloud environment.**
8.	**Implementation Software as a service (SaaS) in private cloud environment.**
9.	**Implementing Software as a service (SaaS) in public cloud environment.**
10.	**Implementation of Storage as a service (SaaS).**

Course: BTech

Prerequisite: Basic knowledge of Blockchain.

303105369 - Hyper ledger & Fabric Programing

Rationale : This course provides a broad introduction to Hyperledger. Hyperledger Fabric is an open-source framework that uses a variety of programming and consensus mechanisms to support private and permissioned business networks:

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction:** Blockchain, Public and Private Blockchain, Enterprise Blockchain, Distributed ledger technology (DLT), Challenges in DLT, Hyperledger, Hyperledger Blockchain Architecture	5	2
2	**Hyperledger Framework:** Enterprise Blockchain, Enterprise Design Issues, Hyperledger, Hyperledger Sawtooth, Hyperledger Iroha, Hyperledger Indy, Hyperledger Burrows, Hyperledger Fabric, Fabric Usecases, Fabric Architecture	10	5
3	**Hyperledger Fabric and Composer:** Hyperledger Fabric for business applications, Assets, ledgers, transactions, and the chaincode, Permissioned network, Services for members and memberships, Channels and nodes	20	10
4	**Smart Contacts in Hyperledger:** Creating Contract, Access Control, Implementing Contract functions, Testing a Contract, Endorcement Policy, Querying the Ledger, Events and Notifications, Go Language	25	12
5	**Setting up a network:** Network Lifecycle, Operations- Network Setup and Bootstrap, Developing Service Layer Applications	20	8
6	**Hyperledger Fabric Security:** Network bootstrap and data governance: the first step toward security, Bootstrapping the network with known members, Defining the process for sharing data, Defining the data model of the shared data, Mapping data sharing needs Hyperledger-based mechanisms for operational agility, Adding new members to the network (or channel), Deploying, instantiating, and upgrading smart contracts on peers in the network, Strong identities: the key to the security of the Hyperledger Fabric network, Bootstrapping Fabric CA – Register, Enroll, Revoking identities, Practical considerations in managing users in Fabric CA, Smart contract security, Who can install smart contracts? Smart contract encryption, Attribute-based access control, Common threats and how Hyperledger Fabric mitigates them, Hyperledger Fabric and quantum computing, GDPR consideration	20	8
	Total	100	45

Reference Books

1.	**Blockchain with Hyperledger Fabric Nitin Gaur, luc Desrosiers (TextBook)**
2.	**Mastering Hyperledger Fabric Narendranath Reddy**

Course Outcome

After Learning the Course the students shall be able to:
1. Understand Enterprise blockchain. 2. Implement smart contracts in Hyperledger 3. Contribute to the open-source Hyperledger projects. 4. Understand and Implement Hyperledger cross-industry use cases and projects. 5. Explain business distributed ledger and blockchain technologies.

Course: Btech 303105370 - Hyper ledger & Fabric Programing Laboratory

Prerequisite: Basic knowledge of Blockchain Technology

Rationale : This course provides a broad introduction to Enterprise Blockchain. Hyperledger Fabric is an open-source framework that uses a variety of programming and consensus mechanisms to support private and permissioned business networks.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Seminar Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **T** - Theory, **P** - Practical

Course Outcome

After Learning the Course the students shall be able to:
1. Understand Enterprise blockchain. 2. Implement smart contracts in Hyperledger 3. Contribute to the open-source Hyperledger projects. 4. Understand and Implement Hyperledger cross-industry use cases and projects. 5. Explain business distributed ledger and blockchain technologies.

List of Practical

1.	Docker Setup for Automation
2.	Docker Commands, Docker RUN
3.	Docker Images, Environment Variables
4.	Docker Compose, Registry and Engine
5.	Container Orchestration
6.	Vagrant: to set up virtual machine, Provision Script
7.	Blockchain, Fabric Model, Conceptual Network, Install Hyperledger Fabric
8.	Hyperledger Set-up, Creating Crypto Material, Deploying Network
9.	Smart Contract Design, Using GO Packages, Templates
10.	Overview of GO commands, Creating a project, Declaring Variables
11.	Working with Pointers, Creating constants, Using Iota and Constant Expressions

12.	Creating Arrays, working with slices
13.	Using Maps, Structs, Adding variables to webservice
14.	Creating functions, Loops, Looping over Collection, Panic Function, If and Switch statements
15.	CouchDB, Private Data Collections, Blockchain Explorer, Writing Applications