Diploma Program in
‘Software development’

Duration: 32 weeks (160 hours)

Sr. No.	Name of the Programme	Details About the Programme	Remarks
1.	**Diploma certificate course in 'Software Development'**	**Mission of AIIITS:** The mission of AIIITS is to advance knowledge and educate students in information Technology and other areas that will best serve the nation. **Vision of AIIITS:** To position AIIITS as a premier institute responsive to emerging needs of industry. To produce high skilled graduates and contribute towards sustainable development of the industry and nation.	
		1. <u>Introduction to the Programme:</u> Software development is the procedure of designing and developing software applications using certain set of instructions and a specific programming language or a framework. The developed application further will be able to perform certain set of tasks depending upon the set of processes used in development. A typical software development actually went through various steps include ideation, specification, designing, programming, documentation, testing, bug fixing, and deployment. Diploma in software development is the comprehensive course which is very trending nowadays among the students who wish to make their career as a developer. Diploma in software development is the most admirable training course today which gives in-depth knowledge of software development concepts and you will be able to design and develop low level to high level software applications. The course aims at utilizing fully the capabilities of the free and open source software. In the training 30% will be theoretical and 70% will be hands-on training. **2. Objectives of the Programme:** The course aims at covering the fundamental of Information Technology and its different aspects, developing logics and algorithms, imparting relevant programming abilities, develop capabilities to manage database and generate queries to extract details from a dataset. It also equips the participants to develop websites, console based software and web based software. **<u>Objectives:</u> the objectives of the proposed course are to impart knowledge on the following:**	

		1. Building concepts of IT infrastructure. 2. Developing logics and algorithms and programming skills to develop software and its connectivity with database. 3. Developing skills to develop websites. 4. Learn skills which can give a very promising career.	
		3. Target Group of Learners: Diploma in software development is meant for anyone who want to make their career in IT and software development. Career options will be there for website development as a database engineer and as a software developer. Any students after completion of this program can get placed in industry as website developer or database engineer or software developer and with experience they can grow in their career path. It also helps to build a promising career in every aspects.	
		4. Instructional Design: The course will consist of live, videos and assignments for every modules. After every module doubt-clearing sessions will be arranged where students will be free to clear their doubts. After the completion of six modules projects will be allotted to students.	
		5. Instructional Design: **i.** **<u>Duration of the Programme:</u> 8** months **ii.** **<u>Course delivery</u>** The course will be entirely delivered ***<u>online or Offiline.</u>*** There are six modules in the course which will be conducted online by expert faculties in the respective areas. Each week's menu will cover the following: **1. <u>Interactive Online lectures:</u>** These sessions will be conducted in classroom or on Microsoft Team or Google Meet or Zoom in case situation arises. The session links will be shared with students. **2. <u>Lecture(s):</u>** the theoretical and applied parts of the topic will be covered in lectures. **3. <u>Exercises and data:</u>** Assignments will be allocated to the participants which they need to complete and submit and assessments on any particular module will be done based on the assignments.	

		4. **Doubt-clearing:** There will be an interactive forum as a platform to interact with each other and with the resource persons**.** Here the participants can discuss their difficulties, can ask questions and get the doubts clarified.	
		6. **Eligibility:** Students who have passed 10+2, any bachelor's in Science/ Arts/ Commerce/ Engineering/ from any recognized universities in India or other countries.	
		7. **Scheme and Evaluation:** There would be three types of assessment for evaluating the performance of the participants - short and long answer questions, multiple type questions and practical exercises. Each participant will be given assignments and projects. After completion of the training, online examination will be conducted and Certificate will be jointly issued by Pinnacle Infotrain and Jain University only after completion of all the assignments, project and after qualifying the exam.	
		8. **Procedure for admission, Curriculum transaction and evaluation:** Admission will be based on prerequisite degree of any recognized universities in India. The Course is affiliated to the Jain University.	
		9. **Fee structure:** The fees should be paid in one installment only, before the commencement of the course*. Rs. 58,000/- *Installment options can be provided with additional 8% on the course fees. The students can pay it in 4 easy installments	
		10. Syllabus: **Topic 1: IT Fundamentals and Concepts -** Introduction to Computer Fundamentals, Components of a PC and their functions, Number system, Operating System, Internetworking and Network, Programming Logic and Techniques, Introduction to Computers, Software Engineering, Problem Solving, Overview of Programming Languages, Procedure oriented Programming, Object oriented Programming, Structuring of a Program, Introduction to data Management concepts, Database Management Systems (DBMS), Quality and Characteristics, Introduction to Basic Database Concepts, E-R Modelling and Diagram, Normalization	

		(Duration-1 Weeks) **Topic 2: C/C++-** Programming in C, Data types, Operators, Control Structures, Loop Control Structure, Arrays, Functions, Library Functions, Recursion, Structures in C, Structure and Arrays, Structure and Functions, Pointers in C, Function Pointer, Macros and Storage Class, File Inclusion, Storage Classes, Files Input/output, Console I/O and File I/O, Command line arguments, Creating your own command, Data Structure, Stacks and Queues, Types of programming language, Compiler and Interpreter, Introduction to OOP, Characteristics of OOP, Advantages of OOP, Introduction to C++, Function overloading, Classes and Objects, Scope resolution operator, Constructors and Destructors, Operator Overloading, Overloading the + operator, Overloading the == operator, Friend functions / classes, Overloading <<and >>, Dynamic Memory Management, Miscellaneous class issues, The const data member, The static data members, Inheritance, Virtual Base Classes, Dynamic Binding, Polymorphism, Upcasted pointer, Size of an object, VTABLE, VPTR, C++ I/O formatting and File Handling cin, cout revisited • manipulators, Linked List Core **(Duration- 5 weeks)** **Topic 3: Introduction to RDBMS -** Types of data models, Entity - relationship model, Entity - relationship diagram to tables, Normalization, Introduction to SQL, SQL server tools, Querying data from a single tables, Querying data from multiple tables, Managing databases and tables, Manipulate data in tables, Views and Indexes, Store procedures and functions, Triggers and transactions, Managed code implementation, Services for message based communication **(Duration-2 weeks)** **Topic 4: Web development-** **HTML-** Introduction, Understanding & using HTML, HTML headings, HTML Comment, HTML Paragraphs, HTML Line Breaks & Rules, HTML Tags, HTML Tables, HTML Nesting, HTML Forms **DHTML-** Introduction to DHTML, DHTML JavaScript, DHTML, HTML DOM, DHTML Events, DHTML CSS, DHTML Examples	

		CSS- What is CSS? CSS Comments, Three Ways to Insert CSS, Multiple Style Sheets, CSS properties used for background effects, Text, CSS Font Families, CSS Links, CSS Lists, Table Borders, CSS Padding **JavaScript-** Introduction to JavaScript, Statements and Variables, Operators, Conditional Statements, Popup Boxes, Arrays, Events and Functions, JavaScript Form, Validation, JavaScript **(Duration-5 weeks)** **<u>Topic 5:</u> Programming Language-** **Java-** Fundamentals of Java Programming, Introduction to java, Primitive Data Types and Operations, Operators in Java, Control Statements, Input and Output, Methods in Java, Arrays, String &String Buffer Object-Oriented, Programming, Introduction OOPs, Object and Classes, Class Inheritance and Interfaces, Polymorphism, Encapsulation, Abstract class, Garbage Collection **(Duration: 3 weeks)s** **<u>Topic 6:</u> DotNet Professional Track-** **C#-** Building C# Applications Using csc.exe, Building .NET Applications Using Notepad++, Building .NET Applications Using SharpDevelop, Building .NET Applications Using Visual C# 2010 Express, Building .NET Applications Using Visual Studio 2010, Core C# Programming Constructs, The Anatomy of a Simple C# Program, An Interesting Aside: Some Additional Members of the System. Environment Class, The System. Console Class, System Data Types and C# Shorthand Notation, Working with String Data, Narrowing and Widening Data Type Conversions, Understanding Implicitly Typed Local Variables, C# Iteration Constructs Decision Constructs and the Relational/Equality Operators, Methods and Parameter Modifiers, Understanding C# Arrays, Understanding the Enum Type, Understanding the Structure Type, Understanding Value Types and Reference Types, Understanding C# Nullable Types, Defining Encapsulated Class Types, Introducing the C# Class Type, Understanding Constructors, The Role of the this Keyword, Understanding the static Keyword, Defining the Pillars of OOP, C# Access Modifiers, The First Pillar: C#'s Encapsulation Services, Understanding Automatic Properties, Understanding Object Initializer	

		Syntax, Working with Constant Field Data, Understanding Partial Types, Understanding Inheritance and Polymorphism The Basic Mechanics of Inheritance, Revising Visual Studio Class Diagrams, The Second Pillar of OOP: The Details of Inheritance, Programming for Containment/Delegation, The Third Pillar of OOP: C#'s Polymorphic Support, Understanding Base Class/Derived Class Casting Rules, The Master Parent Class: System. Object, Understanding Structured Exception Handling, Ode to Errors, Bugs, and Exceptions, The Role of .NET Exception Handling, The Simplest Possible Example, Configuring the State of an Exception, System-Level Exceptions (System. System Exception), Application-Level Exceptions (System. Application Exception), Processing Multiple Exceptions, The Result of Unhandled Exceptions, Debugging Unhandled Exceptions Using Visual Studio, Understanding Object Lifetime, Classes, Objects, and References, The Basics of Object Lifetime, The Role of Application Roots, Understanding Object Generations, Building Finalizable Objects, Building Disposable Objects, Building Finalizable and Disposable Types, Understanding Lazy Object Instantiation, Working with Interfaces, Understanding Interface Types, Defining Custom Interfaces, Implementing an Interface, Invoking Interface Members at the Object Level, Interfaces As Parameters, Interfaces As Return Values, Arrays of Interface Types, Resolving Name Clashes via Explicit Interface Implementation, Designing Interface Hierarchies, Understanding Generics, The Issues with Non-Generic Collections, The Role of Generic Type Parameters, The System.Collections.Generic Namespace, Creating Custom Generic Methods, Creating Custom Generic Structures and Classes, Constraining Type Parameters, Delegates, Events, and Lambdas, Understanding the .NET Delegate Type, Defining a Delegate Type in C#, Understanding Delegate Covariance Understanding Generic Delegates, Understanding C# Events, Understanding C# Anonymous Methods, Understanding Lambda Expressions, Advanced C# Language Features, Understanding Indexer Methods, Understanding Operator Overloading, Understanding Custom Type Conversions, Understanding Extension Methods, Understanding Partial Methods, Understanding Anonymous Types, Working with Pointer Types LINQ Specific Programming Constructs, Understanding the Role of LINQ, Applying LINQ Queries to Primitive Arrays, Returning the Result of a LINQ Query, Applying LINQ Queries to Collection Objects, Investigating the C# LINQ Query Operators, The Internal Representation of LINQ Query Statements, Programming with .NET Assemblies, Configuring .NET Assemblies, Defining	

		Custom Namespaces, The Role of .NET Assemblies, Understanding the Format of a .NET Assembly, Building and Consuming a Single-File Assembly, Building and Consuming a Multifile Assembly, Understanding Private Assemblies, Understanding Shared Assemblies, Consuming a Shared Assembly, Configuring Shared Assemblies, Understanding Publisher Policy Assemblies, Understanding the <codeBase> Element, The System.Configuration Namespace, Dynamically Loading Assemblies, Reflecting on Shared Assemblies, Understanding Late Binding, Understanding the Role of .NET Attributes, Building Custom Attributes, Assembly-Level (and Module-Level) Attributes **Windows Programming-** The Role of a Windows Process, Interacting with Processes Under the .NET Platform, Understanding .NET Application Domains, Interacting with the Default Application Domain, Creating New Application Domains, Understanding Object Context Boundaries, Summarizing Processes, AppDomains, and Context, Understanding CIL and the Role of Dynamic Assemblies, Reasons for Learning the Grammar of CIL, Examining CIL Directives, Attributes, and Opcodes, Pushing and Popping: The Stack-Based Nature of CIL, Understanding Round-Trip Engineering, Understanding CIL Directives and Attribute, The Role of WCF, Investigating the Core WCF Assemblies, The Visual Studio WCF Project Templates, The Basic Composition of a WCF Application, The ABCs of WCF, Building a WCF Service, Hosting the WCF Service, Building the WCF Client Application, Simplifying Configuration Settings with WCF 4.0, Using the WCF Service Library Project Template, Hosting the WCF Service within a Windows Service, Invoking a Service Asynchronously from the Client, Designing WCF Data Contracts, Introducing Windows Workflow Foundation, Building Desktop User Interfaces with WPF Introducing Windows Presentation Foundation and XAML, The Motivation Behind WPF, The Various Flavors of WPF, Investigating the WPF Assemblies, Building a WPF Application without XAML, Building a WPF Application using Only XAML, Transforming Markup into a .NET Assembly, Understanding The Syntax of WPF XAML, Building a WPF Application using Code-Behind Files, Building WPF Applications Using Visual Studio xxxx, Programming with WPF Controls, A Survey of the Core WPF Controls, Controlling Content Layout Using Panels, Building a Window's Frame Using Nested Panels, Understanding WPF Control Commands, Building a WPF User Interface with Expression Blend, Building the Ink API Tab, Introducing the Documents API, Building the Documents	

		Tab, Introducing the WPF Data-Binding Model, WPF Graphics Rendering Services, Understanding WPF's Graphical Rendering Services, Rendering Graphical Data Using Shapes, WPF Brushes and Pens **ASP.NET-** Understanding Web Applications and Web Servers, The Role of HTML, The Role of Client-Side Scripting, Posting Back to the Web Server, The Feature Set of the ASP.NET API, Building a Single File ASP.NET Web Page, Building an ASP.NET Web Page using Code Files, ASP.NET Web Sites and ASP.NET Web Applications, The ASP.NET Web Site Directory Structure, The Inheritance Chain of the Page Type, Interacting with the Incoming HTTP Request, Interacting with the Outgoing HTTP Response, The Life Cycle of an ASP.NET Web Page, The Role of the Web.config File, ASP.NET Web Controls, Master Pages and Themes, Understanding the Nature of Web Controls, The Control and WebControl Base Classes, Major Categories of ASP.NET Web Controls, Building the ASP.NET Cars Web Site, The Role of the Validation Controls, Working with Themes, ASP.NET State Management Techniques, The Issue of State, ASP.NET State Management Techniques, Understanding the Role of ASP.NET View State, The Role of the Global.asax File, Understanding the Application/Session Distinction, Working with the Application Cache, Maintaining Session Data, Understanding Cookies, The Role of the <sessionState> Element, Understanding the ASP.NET Profile API **(Duration-12 weeks)** **Topic 7: Angular-x-** Components, Using Custom Components, Creating Components with the CLI & Nesting Components, Working with Component Templates, Working with Component Style, Practicing Components, What is Databinding, String Interpolation, Property Binding, Property Binding vs String Interpolation, Event Binding, Bindable Properties and Events, Passing and Using Data with Event Binding, Two-Way-Databinding, Important: FormsModule is Required for Two-Way-Binding, Combining all Forms of Databinding, Practicing Databinding, Understanding Directives, Using ngIf to Output Data Conditionally, Enhancing ngIf with an Else Condition, Components & Databinding Deep Dive, Splitting Apps into Components, Property & Event Binding Overview, Binding to Custom Properties, Assigning an Alias to Custom Properties, Binding to Custom Events, Assigning an Alias to Custom Events,	

Custom Property and Event Binding Summary, Understanding View Encapsulation, More on View Encapsulation, Using Local References in Templates, Getting Access to the Template & DOM with @ViewChild, Understanding the Component Lifecycle Lifecycle Hooks, Practicing Property & Event Binding and View Encapsulation, Directives Deep Dive, ngFor and ngIf, ngClass and ngStyle, How to create a Basic Directive, Using Services & Dependency Injection, Why would you Need Services? Creating a Logging Service, Injecting the Logging Service into Components, Creating a Data Service, Understanding the Hierarchical Injector, How many Instances of Service Should It Be? Injecting Services into Services, Using Services for Cross-Component Communication

(Duration- 4 weeks)

Key Reference Books

1. Fundamentals of Information Technology by Deepak Bharihoke.
2. C/C++ Programmer's reference by Herbert Schildt
3. Php: the complete reference by Steven Holzner
4. SQL the complete reference, 3rd Edition by James R Groff and Paul N Weinberg
5. Fundamentals of Database systems by Ramez Elmasri
6. Ng-book: The complete guide to Angular by Felipe Coury, Ari Lerner, Carlos Taborda

11. Quality Assurance:

IQAC (Internal Quality Assurance Cell) is in place to oversee the Programme delivery mechanism and suggest changes specific to industry requirements.

The quality of the programme will be ensured through strict monitoring by an executive committee including the Co-ordinator of the programme, the subject experts, Director. The Co-ordinator of the programme shall ensure the regular student feedback of courses, teachers and programme in the prescribed format towards the end of the semester and the same shall be analyzed to draw conclusions for effecting improvement.

Periodical review meetings on the programme efficacy will be held in which the remarks of teachers on curriculum, syllabi and methods of teaching and

		evaluation will be given due importance. Moreover, the progress and the quality of the programme will be monitored by the Internal Quality Assurance Cell of Pinnacle Infotrain from the outcome and feedback of the learners as well as the proper documentation maintained in the Centre.	
		12. SLM: Self-Learning Material is available in English	