Full Stack Development Course Curriculum by 10Pie

👉 Whether you want to start your IT career or looking for guidance on how to plan your future career path, 10Pie is your learning hub for tech career knowledge.

Here’s a Full Stack Development course syllabus at a glance:

SL No.	Module Name	Topics Covered	Projects
1	**Introduction to Full Stack Development**	Overview of Full Stack Development,, Software development lifecycle (SDLC), Development Environments, Version control with Git and GitHub, Agile development methodologies, Scrum framework	- Environment Setup: Set up an IDE and configure Git. - SDLC Presentation: Outline the software development lifecycle.
2	**Front-End Development Foundations**	HTML5, Semantic HTML and accessibility, CSS3, Advanced selectors and specificity, CSS Grid and Flexbox layout techniques, CSS transitions and animations, Responsive design principles and mobile-first design, Advanced JavaScript, DOM manipulation and event delegation	- Responsive Web Page: Create a responsive webpage. - JavaScript Quiz App: Build a quiz application.
3	**Front-End Frameworks and Libraries**	React.js, Component architecture and lifecycle methods, State management with Context API and Redux, React Router for navigation and routing, Hooks (useState, useEffect, custom hooks), Angular, Component-based architecture and data binding, Dependency injection and services, Angular routing and guards, Reactive programming with RxJS and observables	- Single Page Application: Develop an SPA with React.js. - Task Management App: Create a task manager using Angular or Vue.js.
4	**Back-End Development**	Node.js, Event-driven architecture and non-blocking I/O, Node Package Manager (NPM) and package management, Building RESTful APIs with Express.js, Express.js, Middleware functions and error handling, Routing and dynamic route parameters, API versioning and documentation (Swagger/OpenAPI), CORS and security best practices	- RESTful API: Create an API for CRUD operations. - Authentication System: Implement user authentication.

5	**Database Management**	SQL Databases, Database design principles (normalization and denormalization), CRUD operations using SQL (PostgreSQL, MySQL), Using ORMs (Sequelize, TypeORM), NoSQL Databases, Introduction to MongoDB and document-oriented databases, Data modeling with Mongoose, Aggregation framework, indexing, and performance optimization	- Database Integration: Connect a web app to a SQL database. - MongoDB CRUD App: Build an app using MongoDB.
6	**Authentication and Security**	User Authentication, Implementing JWT (JSON Web Tokens) for stateless authentication, OAuth 2.0 and OpenID Connect, Password hashing and salting with bcrypt, Application Security, Understanding OWASP Top 10 vulnerabilities, Input validation and sanitization techniques, HTTPS, SSL/TLS	- Secure Login System: Implement JWT authentication. - Vulnerability Assessment: Analyze security risks in an app.
7	**DevOps and Deployment**	Containerization with Docker, Building and managing Docker containers, Docker Compose for multi-container applications, Best practices for Dockerfile, Continuous Integration and Deployment (CI/CD), Setting up CI/CD pipelines using tools (Jenkins, GitHub Actions), Automated testing (unit, integration, end-to-end), Cloud Hosting and Server Management, Deploying applications on cloud platforms (AWS, Azure, GCP)	- Dockerized Application: Containerize a web app. - CI/CD Pipeline: Set up automated deployment.
8	**Advanced Topics**	Progressive Web Apps (PWAs), Service workers and caching strategies, Web App Manifest and installation prompts, Push notifications and background sync, Real-Time Applications, Implementing WebSockets and Socket.IO for real-time communication, Microservices Architecture, Principles of microservices and service-oriented architecture (SOA), API Gateway patterns and service discovery	- PWA Development: Build an offline-capable web app. - Microservices Project: Create a microservices application.
9	**Testing and Quality Assurance**	Testing Strategies, Unit testing with Jest and Mocha, Integration testing, End-to-end testing with Cypress or Selenium, Test-driven development (TDD) principles, Code Quality and Static Analysis, Linting and formatting tools (ESLint, Prettier), Code reviews and best practices	- Automated Testing Suite: Create tests using Jest and Cypress. - Code Review Practice: Conduct a peer code review.

Module 1: Introduction to Full-Stack Development

Overview of Full Stack Development

- Roles of a Full Stack Developer
- Software development lifecycle (SDLC)

Development Environments

- Setting up development environments (IDE, text editors)
- Version control with Git and GitHub

Agile development methodologies

Scrum framework

⭐ Hands-on projects to practice:

- **Environment Setup Project:** Set up a development environment using a chosen IDE (e.g., VSCode) and configure Git for version control.
- **SDLC Presentation:** Create a presentation that outlines the software development lifecycle, including phases and methodologies.

Module 2: Front-End Development Foundations

HTML5

- Semantic HTML and accessibility (ARIA)
- Forms, validations, and custom data attributes
- Multimedia elements (audio, video)

CSS3

- Advanced selectors and specificity
- CSS Grid and Flexbox layout techniques
- CSS transitions and animations
- Responsive design principles and mobile-first design

Advanced CSS technique

- CSS variables
- CSS Grid advanced features

JavaScript

- Advanced JavaScript concepts (closures, prototypes)
- Asynchronous JavaScript (callbacks, promises, async/await)

- DOM manipulation and event delegation
- Error handling and debugging techniques

⭐ **Hands-on projects to practice:**

- **Responsive Web Page:** Build a responsive webpage using HTML5 and CSS3 that includes forms, and multimedia elements, and follows accessibility standards.
- **JavaScript Quiz App:** Create a quiz application that uses JavaScript for DOM manipulation, event handling, and asynchronous operations.

Module 3: Front-End Frameworks and Libraries

React.js

- Component architecture and lifecycle methods
- State management with Context API and Redux
- React Router for navigation and routing
- Hooks (useState, useEffect, custom hooks)

Angular

- Component-based architecture and data binding
- Dependency injection and services
- Angular routing and guards
- Reactive programming with RxJS and observables

Vue.js

- Vue instance and lifecycle hooks
- Vue Router for SPA navigation
- Vuex for state management
- Custom directives and mixins

⭐ **Hands-on projects to practice:**

- **Single Page Application (SPA):** Develop an SPA using React.js that fetches data from an API and displays it with routing and state management.
- **Task Management App:** Build a task management application using Angular or Vue.js that allows users to create, update, and delete tasks.

Module 4: Back-End Development

Node.js

- Event-driven architecture and non-blocking I/O
- Node Package Manager (NPM) and package management

- Building RESTful APIs with Express.js

Express.js

- Middleware functions and error handling
- Routing and dynamic route parameters
- API versioning and documentation (Swagger/OpenAPI)
- CORS and security best practices

⭐ Hands-on projects to practice:

- **RESTful API:** Create a RESTful API using Express.js that performs CRUD operations on a resource (e.g., users or products) with proper routing and middleware.
- **Authentication System:** Develop a simple authentication system using Node.js and Express.js that allows users to register and log in using JWT.

Module 5: Database Management

SQL Databases

- Database design principles (normalization and denormalization)
- CRUD operations using SQL (PostgreSQL, MySQL)
- Using ORMs (Sequelize, TypeORM)

NoSQL Databases

- Introduction to MongoDB and document-oriented databases
- Data modeling with Mongoose
- Aggregation framework, indexing, and performance optimization

Advanced MongoDB topics

- MongoDB Atlas
- MongoDB Stitch

⭐ Hands-on projects to practice:

- **Database Integration Project:** Create a web application that integrates with an SQL database (PostgreSQL or MySQL) to manage user data.
- **MongoDB CRUD Application:** Build a full-stack application that uses MongoDB for data storage, allowing users to perform CRUD operations through a web interface.

Module 6: Authentication and Security

User Authentication

- Implementing JWT (JSON Web Tokens) for stateless authentication
- OAuth 2.0 and OpenID Connect
- Password hashing and salting with bcrypt

Application Security

- Understanding OWASP Top 10 vulnerabilities
- Input validation and sanitization techniques
- HTTPS, SSL/TLS
- Secure web practices(secure coding practices, threat modeling)

⭐ **Hands-on projects to practice:**

- **Secure Login System:** Implement a secure login system in a web application that uses JWT for authentication and includes password hashing.
- **Vulnerability Assessment:** Conduct a vulnerability assessment on an existing application and create a report on potential security risks and mitigations.

Module 7: DevOps and Deployment

Containerization with Docker

- Building and managing Docker containers
- Docker Compose for multi-container applications
- Best practices for Dockerfile
- Docker Swarm, Docker Compose
- Kubernetes and container orchestration

Continuous Integration and Deployment (CI/CD)

- Setting up CI/CD pipelines using tools (Jenkins, GitHub Actions)
- Automated testing (unit, integration, end-to-end)

Cloud Hosting and Server Management

- Deploying applications on cloud platforms (AWS, Azure, GCP)
- Load balancing, scaling, and monitoring applications
- Serverless architecture with AWS Lambda and API Gateway

⭐ **Hands-on projects to practice:**

- **Dockerized Application:** Containerize a web application using Docker and create a Docker Compose file to manage multi-container applications.
- **CI/CD Pipeline:** Set up a CI/CD pipeline using GitHub Actions to automate testing and deployment of a web application to a cloud platform (e.g., Heroku or AWS).

Module 8: Advanced Topics

Progressive Web Apps (PWAs)

- Service workers and caching strategies
- Web App Manifest and installation prompts
- Push notifications and background sync
- PWA monetization

Real-Time Applications

- Implementing WebSockets and Socket.IO for real-time communication
- Pub/Sub messaging patterns (Redis, RabbitMQ)

Microservices Architecture

- Principles of microservices and service-oriented architecture (SOA)
- API Gateway patterns and service discovery
- Inter-service communication (REST, gRPC)

⭐ Hands-on projects to practice:

- **PWA Development:** Build a Progressive Web App that works offline and includes service workers for caching and push notifications.
- **Microservices Project:** Create a microservices-based application where different services handle specific functionalities (e.g., user service, product service) and communicate via REST or gRPC.

Module 9: Testing and Quality Assurance

Testing Strategies

- Unit testing with Jest and Mocha
- Integration testing
- End-to-end testing with Cypress or Selenium
- Test-driven development (TDD) principles

Code Quality and Static Analysis

- Linting and formatting tools (ESLint, Prettier)
- Code reviews and best practices

Continuous Monitoring and Feedback

⭐ Hands-on projects to practice:

- **Automated Testing Suite:** Develop an automated testing suite for an existing application using Jest and Cypress to ensure functionality and performance.
- **Code Review Practice:** Conduct a code review session with peers, providing feedback on code quality and suggesting improvements based on best practices.

B.Sc in Software Engineering

Focuses on software development methodologies and practices, including both front-end and back-end development skills.

Typically, a B.Sc (Bachelor of Science) course in Software Engineering is a 3-year program with 6 semesters. The average value in India is around ₹1,75,000 per year. Here is a quick glance of the curriculum:

Semester	Subject	Topic
1	**Introduction to Computer Systems**	Computer architecture, operating systems, input/output systems, memory management, and computer networks
	Programming Fundamentals	Introduction to programming, data types, variables, control structures, functions, and object-oriented programming
	Discrete Mathematics	Set theory, relations, functions, graph theory, and combinatorics
2	**Data Structures and Algorithms**	Arrays, linked lists, stacks, queues, trees, graphs, and algorithm design techniques
	Object-Oriented Programming	Object-oriented programming concepts, inheritance, polymorphism, and encapsulation
	Computer Organization and Architecture	CPU architecture, memory hierarchy, input/output systems, and computer networks
3	**Software Engineering Fundamentals**	Software engineering principles, software development life cycle, and software project management
	Web Development	HTML, CSS, JavaScript, and web development frameworks
	Database Systems	Database concepts, data modeling, database design, and database management systems
4	**Operating Systems**	Operating system concepts, process management, memory management, and file systems
	Computer Networks	Computer network fundamentals, network protocols, and network architecture

	Software Design and Architecture	Software design principles, design patterns, and software architecture
5	**Artificial Intelligence and Machine Learning**	Artificial intelligence principles, machine learning algorithms, and applications
	Data Mining and Business Intelligence	Data mining concepts, data warehousing, and business intelligence
	Cloud Computing	Cloud computing concepts, cloud infrastructure, and cloud applications
6	**Software Project Management**	Software project management principles, project planning, and project execution
	Data Science	Data science principles, data visualization, and data analytics
	Software Engineering Capstone Project	Software engineering project development, project management, and project presentation

Bachelor's Degree in Computer Science:

These 4-year degree programs provide a strong foundation in programming, algorithms, databases, and software development, which are essential for full stack development.

Courses cover both front-end (HTML, CSS, JavaScript) and back-end (programming languages, APIs, databases) technologies.

The average range of B.Tech Computer Science course fees in India is around ₹1,50,000 - ₹2,50,000 per year. However, the fees can vary depending on the institution, location, and other factors.

Semester	Course Title	Topic
1	**Programming Fundamentals**	Introduction to programming, data types (primitive and composite), variables, control structures (conditional statements, loops, and jumps), functions, and arrays (1D and 2D)
	Discrete Mathematics	Set theory (operations, laws, and properties), relations (types and properties), functions (types and properties), graph theory (basic concepts and graph traversals), and combinatorics (permutations and combinations)

	Computer Organization	CPU architecture (fetch-decode-execute cycle, registers, and ALU), memory hierarchy (main memory, cache, and virtual memory), and input/output systems (types and interfaces)
2	**Data Structures and Algorithms**	Arrays (operations and applications), linked lists (singly and doubly linked), stacks (operations and applications), queues (operations and applications), trees (basic concepts and traversals), and graphs (basic concepts and traversals)
	Computer Networks	Network fundamentals (ISO/OSI model, TCP/IP model), protocols (HTTP, FTP, and DNS), and network architecture (LAN, WAN, and Wi-Fi)
	Operating Systems	Process management (process creation, synchronization, and communication), memory management (memory allocation and deallocation), and file systems (file organization and management)
3	**Database Systems**	Database design (ER diagrams, normalization), data modeling (data warehousing and data mining), and query languages (SQL and NoSQL)
	Software Engineering	Software development life cycle (SDLC models), design patterns (creational, structural, and behavioral), and testing (types and techniques)
	Web Development	HTML (structure and syntax), CSS (selectors, properties, and values), JavaScript (syntax, DOM, and events), and web frameworks (React, Angular, and Vue)
4	**Artificial Intelligence**	Introduction to AI (history, types, and applications), machine learning (supervised, unsupervised, and reinforcement learning), and neural networks (basic concepts and architectures)
	Computer Graphics	Graphics fundamentals (2D and 3D graphics, graphics pipeline), 2D graphics (transformations, clipping, and rendering), and 3D graphics (projections, lighting, and rendering)
	Human-Computer Interaction	User experience design (principles, process, and methods), human factors (cognitive psychology, ergonomics), and usability testing (methods and metrics)
5	**Data Mining and Warehousing**	Data preprocessing (data cleaning, data transformation), data mining techniques (classification, clustering, and association rule mining), and data warehousing (architecture, design, and implementation)
	Network Security	Network security threats (types and examples), vulnerabilities (types and examples), and security measures (firewalls, VPNs, and access control)

	Computer Vision	Image processing (image filtering, image segmentation), object recognition (feature extraction, object detection), and computer vision applications (image classification, object tracking)
6	**Distributed Systems**	Distributed system architecture (client-server, peer-to-peer), communication protocols (TCP/IP, HTTP), and synchronization (process synchronization, data synchronization)
	Information Systems	Information system design (system analysis, system design), development (system implementation, system testing), and implementation (system deployment, system maintenance)
7	**Advanced Computer Networks**	Advanced network topics (network simulation, network modeling, network performance analysis)
	Advanced Operating Systems	Advanced operating system topics (distributed operating systems, real-time systems, embedded systems)
8	**Capstone Project**	Students work on a project that integrates knowledge and skills acquired throughout the program
	Elective Course	Students choose an elective course from a list of available courses (Advanced AI, Advanced Computer Graphics, Advanced Network Security, etc.)

Master's Degree in Computer Science or Software Engineering

Advanced degree programs that provide deeper knowledge of software architecture, design patterns, and emerging technologies ideal for those who want to specialize in full-stack development.

Master of Science in Computer Science (MSCS)

Semester	Course	Description
1	**Foundations of Computer Science**	Discrete Mathematics: Set theory, combinatorics, and graph theory. Data Structures and Algorithms: Arrays, linked lists, stacks, queues,

		trees, and graphs. Computer Organization and Architecture: CPU architecture, memory hierarchy, and input/output systems.
	Programming Fundamentals	Programming Languages: Syntax, semantics, and pragmatics of programming languages (e.g., Python, Java, C++). Data Analysis and Visualization: Data preprocessing, data visualization, and statistical analysis.
2	**Advanced Computer Science Topics**	Operating Systems: Process management, memory management, file systems, and security. Computer Networks: Network fundamentals, protocols, network architecture, and network security. Database Systems: Database design, data modeling, database management systems, and data warehousing.
	Software Engineering	Software Design and Development: Software design patterns, principles, and methodologies. Software Testing and Quality Assurance: Testing techniques, test-driven development, and quality assurance.
3	**Specialized Computer Science Topics**	Artificial Intelligence and Machine Learning: Introduction to AI, machine learning, and deep learning. Data Science and Analytics: Data visualization, data mining, and data analytics. Cybersecurity: Network security, cryptography, and cybersecurity principles.
	Research Methods and Tools	Research Methodologies: Research design, data collection, and data analysis. LaTeX and Technical Writing: LaTeX basics, technical writing, and research paper writing.
4	**Thesis or Project**	Original research or project under the guidance of a faculty advisor.

Master of Science in Software Engineering (MSSE)

Semester	Course	Description
1	**Foundations of Software Engineering**	Software Engineering Principles and Practices: Software development life cycles, software project management, and software engineering principles. Software Development Life Cycles: Agile, waterfall, and iterative development methodologies. Software Project Management: Project planning, project scheduling, and project monitoring.
	Programming Fundamentals	Programming Languages: Syntax, semantics, and pragmatics of programming languages (e.g., Python, Java, C++). Data Analysis and Visualization: Data preprocessing, data visualization, and statistical analysis.

2	**Software Design and Development**	Software Design Patterns and Principles: Creational, structural, and behavioral design patterns. Software Development Methodologies: Agile, waterfall, and iterative development methodologies. Software Testing and Quality Assurance: Testing techniques, test-driven development, and quality assurance.
	Software Engineering Tools and Technologies	Version Control Systems: Git and other version control systems. Integrated Development Environments: Eclipse, Visual Studio, and other IDEs. Cloud Computing Platforms: AWS, Azure, and other cloud platforms.
3	**Advanced Software Engineering Topics**	Software Architecture and Design: Software architecture patterns, design principles, and architecture evaluation. Software Security and Testing: Software security principles, testing techniques, and security testing. Human-Computer Interaction and User Experience: User experience design, human-computer interaction principles, and interface design.
	Research Methods and Tools	Research Methodologies: Research design, data collection, and data analysis. LaTeX and Technical Writing: LaTeX basics, technical writing, and research paper writing.
4	**Thesis or Project**	Original research or project under the guidance of a faculty advisor.

M.Tech in Computer Science and Engineering

M.Tech in Computer Science with a specialization in Web and Mobile Applications focuses on the design, development, and deployment of web and mobile applications, which is closely related to full-stack development.

Semester	Course Title	Technical Topics
1	Mathematical Foundations of CS	Discrete Math: Sets, Relations, Graph Theory; Linear Algebra: Vector Spaces, Matrix Operations; Calculus: Limits, Derivatives, Integrals; Probability: Random Variables, Bayes' Theorem; Statistics: Descriptive Statistics, Inferential Statistics
	Computer Systems and Architecture	Computer Organization: CPU, Memory, Input/Output; Architecture: Instruction Set, Pipelining, Cache Memory; OS Concepts: Process Management, Memory Management, File Systems; Microprocessors: Architecture, Instruction Set, Programming

	Programming Fundamentals	C: Variables, Data Types, Operators, Control Structures; C++: Object-Oriented Programming, Templates, Exception Handling; Data Structures: Arrays, Linked Lists, Stacks, Queues; Algorithms: Sorting, Searching, Graph Traversal; Object-Oriented Programming: Classes, Objects, Inheritance
	Computer Networks	Network Fundamentals: Network Topology, Network Protocols; Network Protocols: TCP/IP, HTTP, FTP; Network Architecture: LAN, WAN, Wi-Fi; TCP/IP: IP Addressing, Routing, Socket Programming
	Software Engineering	SDLC: Waterfall, Agile, V-Model; Software Design: UML, Design Patterns; Testing: Black Box, White Box, Gray Box; Agile Development: Scrum, Kanban, Extreme Programming
2	Data Management Systems	DBMS: Database Design, Database Normalization; Data Modeling: ER Diagrams, Relational Model; Data Normalization: 1NF, 2NF, 3NF; SQL: Query Language, Database Schema; NoSQL: Key-Value, Document-Oriented, Graph Databases
	Computer Vision and Image Processing	Image Processing: Image Filtering, Image Segmentation; Computer Vision: Object Recognition, Image Classification; Object Recognition: Face Recognition, Object Detection; Machine Learning: Supervised, Unsupervised, Reinforcement Learning
	Artificial Intelligence and Machine Learning	AI Fundamentals: Intelligent Agents, Problem Solving; ML Algorithms: Linear Regression, Decision Trees, Random Forest; Neural Networks: Perceptron, Multilayer Perceptron, Backpropagation; Deep Learning: Convolutional Neural Networks, Recurrent Neural Networks
	Web Technologies	HTML: Structure, Tags, Attributes; CSS: Styles, Layout, Responsive Design; JavaScript: Syntax, DOM, Events; Web Development: Client-Server Architecture, RESTful APIs
	Elective 1 (HCI, Computer Graphics, Network Security)	HCI: Human-Computer Interaction, User Experience; Computer Graphics: 2D, 3D Graphics, Animation; Network Security: Cryptography, Firewalls, Intrusion Detection
3	Advanced Computer Networks	Network Security: Cryptography, Access Control; Network Protocols: DNS, DHCP, NAT; Network Architecture: Network Virtualization, Software-Defined Networking; SDN: OpenFlow, Network Controllers
	Distributed Systems	Distributed System Design: Client-Server, Peer-to-Peer; Distributed Algorithms: Mutual Exclusion, Deadlock Detection; Cloud Computing: IaaS, PaaS, SaaS

	Data Mining and Warehousing	Data Mining: Association Rule Mining, Clustering; Data Warehousing: Data Modeling, ETL, OLAP; Business Intelligence: Data Visualization, Reporting; Data Visualization: Charts, Graphs, Heat Maps
	Cloud Computing	Cloud Infrastructure: Virtualization, Containerization; Cloud Architecture: Microservices, Serverless Computing; Cloud Security: Identity and Access Management, Compliance; Cloud Migration: Assessment, Planning, Execution
	Elective 2 (Cybersecurity, IoT, Big Data Analytics)	Cybersecurity: Threat Analysis, Vulnerability Assessment; IoT: Device Management, Data Analytics; Big Data Analytics: Hadoop, Spark, NoSQL Databases
4	Research Methodology and IPR	Research Methodology: Research Design, Data Collection; IPR: Patent Law, Copyright Law, Trademark Law; Patent Law: Patentability, Patent Application; Copyright Law: Copyright Infringement, Fair Use
	Project Work	Project Development: Project Planning, Project Execution; Project Management: Agile, Scrum, Waterfall; Agile Development: Iterations, Sprints, Kanban
	Elective 3 (NLP, Computer Forensics, Embedded Systems)	NLP: Text Processing, Sentiment Analysis; Computer Forensics: Digital Evidence, Incident Response; Embedded Systems: Microcontrollers, Real-Time Systems
	Elective 4 (Data Science, Blockchain Technology, Robotics and Automation)	Data Science: Data Wrangling, Data Visualization; Blockchain Technology: Blockchain Architecture, Smart Contracts, Cryptocurrencies; Robotics and Automation: Robot Operating System, Motion Planning, Computer Vision for Robotics

Full Stack development course subjects and topics to learn

Following are the top five subjects and topics you must learn in a full-stack development course:

Web development fundamentals

There are some key fundamental topics you must learn in web development to become a good full-stack developer.

First, learn the basics of core technologies of web development:

- HTML
- CSS

- JavaScript

Second, learn important visual design elements to create aesthetic interfaces such as:

- Website templates
- Typography
- Colour theory

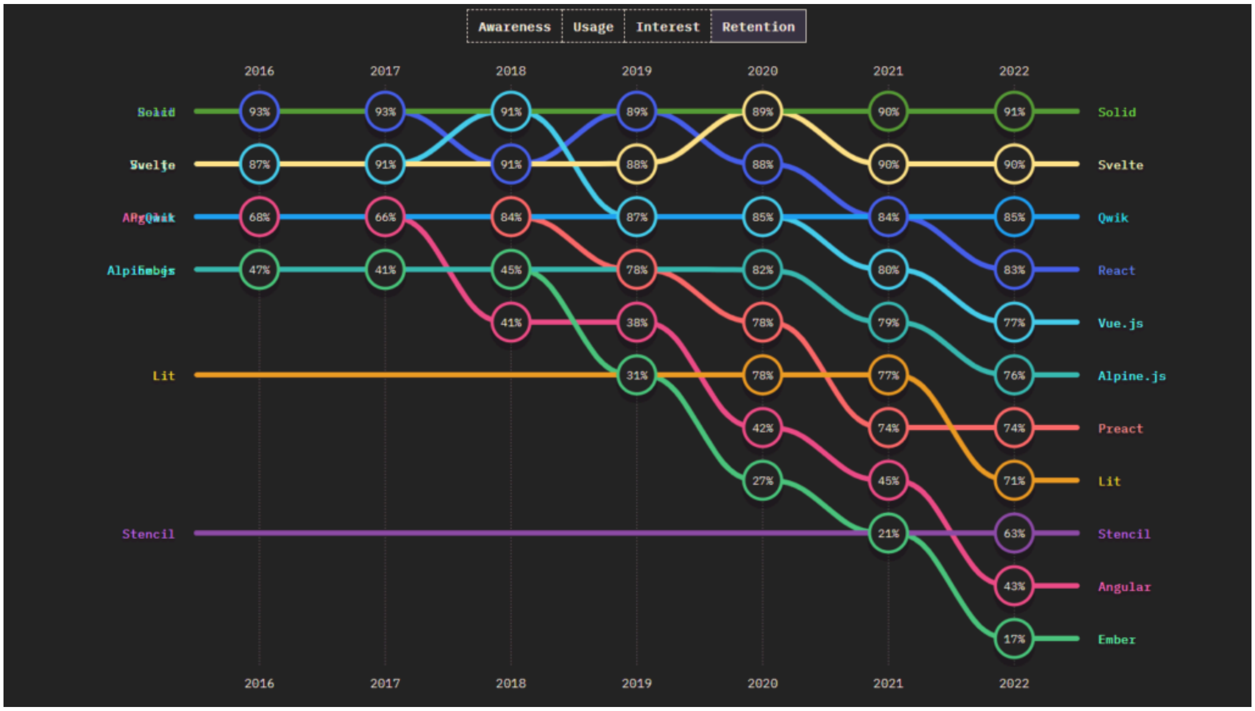
Lastly, have an understanding of version control systems like Git to track code and manage projects efficiently.

Apart from these, it’s also beneficial to be aware of best practices for user accessibility, technical SEO, and graphic design tools.

Front-end frameworks

Front-end frameworks are libraries or tools that simplify the development of UIs for web applications. Learning these frameworks improves you create actual professional-grade applications/projects you can showcase.

As per a report by The State of JS in 2022, these are the following most-used front-end frameworks:



Back-end development

Back-end development is important to manage server-side logic, database management, and applications. Students must learn this to integrate front-end frameworks with back-end services. This ensures a better data flow and communication between the UI and server.

The following are the most-used rendering, or back-end, frameworks as per the same research above:

- Next.js
- Gatsby
- Nuxt
- SvelteKit
- Astro
- Docusaurus
- Remix
- Eleventy

Database management

Database management is the process where you create, maintain, and manipulate databases. This is a fundamental aspect of modern web applications, making it an important topic for students to learn.

Some most common database management tools include MySQL, PostgreSQL, and MongoDB. You can now create simple web applications using front-end frameworks that interact with back-end databases. Make sure to focus on CRUD (create, read, update, delete) operations.

DevOps and deployment

Having knowledge of DevOps helps you create applications that are efficiently built and maintained. This helps you enhance user experiences while ensuring the best application performance.

Hence, DevOps practices streamline the software development lifecycle and bridge the gap between development and operations. These are some key topics:

- CI/CD
- Version control (Git/Github)
- Containzeration, orchestration
- Infrastructure as code
- Monitoring and logging
- Web servers
- Networking and load balancing

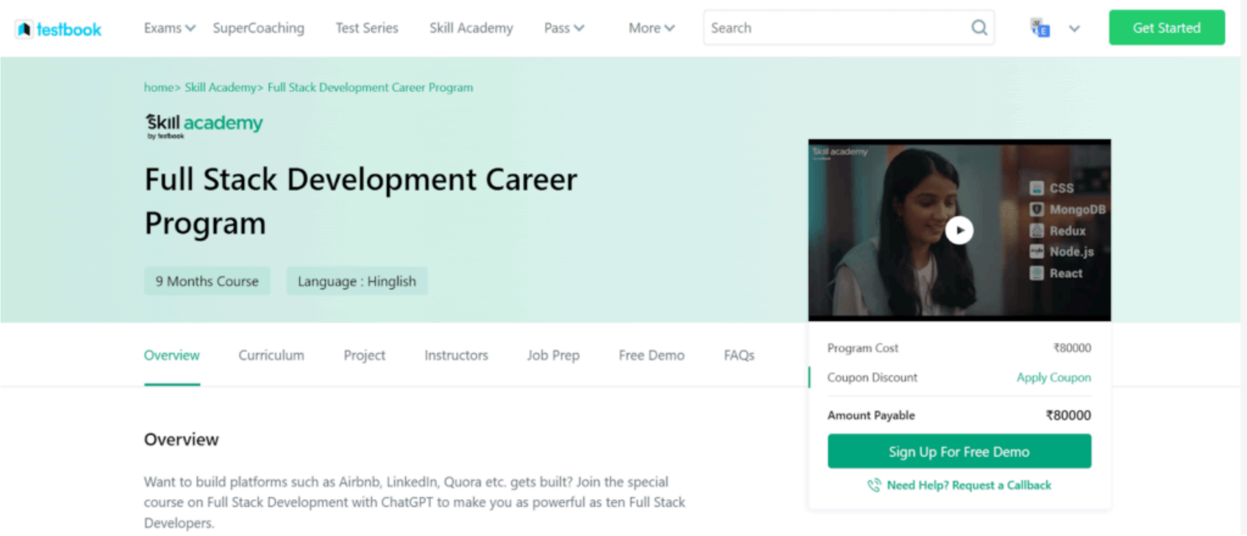
Full Stack development course fees and duration 2024

In 2024, a full-stack developer course covers both front-end and back-end technologies. Its fees and duration vary based on faculty, curriculum, labs, and projects.

What is the course fee for full-stack developer courses?

The fees of a full-stack development course range between ₹20,000 to ₹3,00,000 depending on the program, format, and location. Course fees change depending on the institution, course content, mentorship, and certification.

For instance, Testbook offers a ₹80,000 course that covers all of the above-mentioned topics.



home> Skill Academy> Full Stack Development Career Program

Full Stack Development Career Program

9 Months Course

Language : Hinglish

Program Cost ₹80000

Coupon Discount Apply Coupon

Amount Payable ₹80000

Overview

Want to build platforms such as Airbnb, LinkedIn, Quora etc. gets built? Join the special course on Full Stack Development with ChatGPT to make you as powerful as ten Full Stack Developers.

Full-stack Developer Course Duration

A full-stack developer course lasts 3 months to 12 months. The same Testbook course above offers the course for 9 months. This is another on-par full-stack developer course from Simplilearn, offered for 6 months with a lower price standing at ₹54,000.

When choosing between numerous courses, consider the most important topics discussed above to make an informed decision. Consider factors like a number of real-world projects, accreditation, practical labs, and tools to be learned.

Who is eligible for Full Stack Developer courses?

To enroll in an online full-stack development course, having basic programming, HTML, CSS, and JavaScript knowledge is beneficial.

In India, academic courses have specific eligibility criteria:

- **B.Sc in Full Stack Development:** 12th grade with English and Mathematics/Statistics

- **B.Tech in Computer Science/IT:** 10+2 with Physics, Chemistry, and Mathematics (50% marks)
- **M.Sc in Computer Science:** Bachelor's degree in Computer Science or related field (50% marks)
- **M.Tech in Computer Science:** B.Tech or equivalent degree in a relevant field (50% marks)