

# **SAURASHTRA UNIVERSITY**

**RAJKOT – INDIA**



**Accredited Grade A by NAAC (CGPA 3.05)**

**CURRICULAM**

**FOR**

**M. Sc. (IT & CA)**

**(2 Years Full Time: 4 Semester Programme)**

**MASTER OF SCIENCE**

**(Information Technology & Computer Application)**

**(Semester 3 & 4)**

**Effective From June – 2023**

**M.Sc. (IT & CA)**  
**Saurashtra University**  
**Effective from June - 2023**  
**M.Sc. (IT & CA) (Semester – 3)**

SR. NO.	SUBJECT	No. of LECT./Lab. PER WEEK	CREDIT
1.	<b>CS – 13</b> FLUTTER APP DEVELOPMENT	5	5
2.	<b>CS – 14</b> NODE JS	5	5
3.	<b>CS – 15</b> WEB DEVELOPMENT USING ANGULAR FRAMEWORK	5	5
4.	<b>CS – 16</b> PRACTICAL - 1 (BASED ON CS-13)	5	5
5.	<b>CS – 17</b> PRACTICAL-2 (BASED ON CS-14 and (CS-15 OR CS-16))	5	5
6.	<b>CS – 18</b> PROJECT DEVELOPMENT (In House)	5	5
<b>Total Credits of Semester – 3</b>			<b>30</b>

Note:

1. Total marks of each **theory paper** are 100 (university examination of 70 marks + internal examination of 30 marks).
2. Total marks of each **practical and project-viva** paper are 100. No internal examination of marks in practical and project-viva papers.

**M.Sc. (IT & CA)**  
**Saurashtra University**  
**Effective from June - 2023**

**CS-13: FLUTTER APP DEVELOPMENT**

**Objectives:**

- To understand the basic concepts of Flutter and Dart programming language, including its history and development environment.
- To design and build UIs in Flutter, including stateful widgets and responsive design techniques.
- To manage application state in Flutter using various techniques, such as InheritedWidget and ScopedModel.
- To integrate networking and persistence into their Flutter apps, including working with APIs and local storage.
- To understand advanced Flutter topics such as animations, internationalisation, and platform-specific integration.

**Pre-Requisites:** Basic knowledge of Programming, OOPs Concepts, Knowledge of Native Android Development

Sr. No	Topics	Details	Weightage in %	Approx Lectures
1	<b>Introduction to Flutter and Dart</b>	<ul style="list-style-type: none"> <li>• Overview of Flutter and Dart</li> <li>• Overview of the Flutter architecture and how it works</li> <li>• Setting up the development environment</li> <li>• Dart syntax and data types</li> <li>• Basic Flutter widgets and layout</li> <li>• Basic Flutter widget properties and methods</li> <li>• Dart libraries and packages</li> <li>• Control flow and loops in Dart</li> </ul>	<b>20</b>	<b>12</b>
2	<b>Building User Interfaces</b>	<ul style="list-style-type: none"> <li>• Stateful vs. Stateless widgets</li> <li>• Layout and widgets hierarchy</li> <li>• Navigation and routing</li> <li>• Responsive design and media queries</li> <li>• Advanced Flutter widgets (e.g., SliverAppBar, AnimatedContainer)</li> <li>• Custom widget creation in Flutter</li> <li>• Debugging UI issues in Flutter</li> <li>• Advanced layout techniques (e.g., Flexbox, GridView)</li> </ul>	<b>20</b>	<b>12</b>
3	<b>Managing App State</b>	<ul style="list-style-type: none"> <li>• State management in Flutter</li> <li>• InheritedWidget and InheritedModel</li> <li>• ScopedModel and Provider</li> <li>• BLoC pattern for state management</li> <li>• Stream-based state management with</li> </ul>	<b>20</b>	<b>12</b>

**M.Sc. (IT & CA)**  
**Saurashtra University**  
**Effective from June - 2023**

<b>4</b>	<b>Networking and Persistence</b>	<p>RxDart</p> <ul style="list-style-type: none"> <li>• Redux architecture for Flutter</li> <li>• Firebase integration for state management</li> <li>• Using the Flutter DevTools for debugging</li> </ul> <ul style="list-style-type: none"> <li>• RESTful APIs and HTTP requests</li> <li>• JSON serialization and deserialization</li> <li>• SQLite and local storage</li> <li>• Shared preferences and secure storage</li> <li>• WebSockets for real-time communication in Flutter</li> <li>• Firebase integration for data storage</li> <li>• Caching data in Flutter</li> <li>• Using third-party libraries for networking and data storage (e.g., Dio, Hive)</li> </ul>	<b>20</b>	<b>12</b>
<b>5</b>	<b>Animation, Integration, Testing, Debugging &amp; Accessibility</b>	<ul style="list-style-type: none"> <li>• Animations and motion</li> <li>• Advanced animation techniques (e.g., Flare, Lottie)</li> <li>• Internationalisation and localization</li> <li>• Native platform integration</li> <li>• Push notifications in Flutter (Firebase)</li> <li>• Integration with other native features (e.g., camera, location)</li> <li>• Testing and debugging</li> <li>• Accessibility in Flutter apps</li> </ul>	<b>20</b>	<b>12</b>
		<b>Total</b>	<b>100</b>	<b>60</b>

<b>References Books / URL</b>
1. Flutter in Action, written by Eric Windmill, January 2020, Manning Shelter Island, ISBN 9781617296147
2. Dart Programming for Flutter, written by Carmine Zaccagnino, Feb-2020, Publisher: Pragmatic Bookshelf, ISBN: 9781680506952
3. Flutter Cookbook: Over 100 Proven techniques and Solutions for Development with Flutter 2.2 and Dart, Simone Alessandria, Brian Kayfitz, 2021, Packt Publishing, ISBN 978-1838823382
4. Learning Dart, 2 <sup>nd</sup> Edition, by Ivo Balbaert, Dzenan Ridjanovic, Packt Publishing, ISBN 10: 1785287621
5. Flutter Complete Reference: Create beautiful, fast and native apps for any device, Alberto Miola, Sep-2020
6. <a href="https://fluttercompleterefrence.com/">https://fluttercompleterefrence.com/</a>

7. <https://flutter.dev/>
8. <https://developers.google.com/learn/pathways/intro-to-flutter>

**Course Outcome:**

After completion of the course students will be able:

- Able to understand and Implement the basic concepts of Flutter and Dart programming language, including its history and development environment.
- Able to design and build UIs in Flutter, including stateful widgets and responsive design techniques.
- Able to manage application state in Flutter using various techniques, such as InheritedWidget and ScopedModel.
- Able to integrate networking and persistence into their Flutter apps, including working with APIs and local storage.
- Able to understand advanced Flutter topics such as animations, internationalisation, and platform-specific integration.

**M.Sc. (IT & CA)**  
**Saurashtra University**  
**Effective from June - 2023**

<b>CS – 14: NODE JS</b>				
<b>Objectives:</b>				
<ul style="list-style-type: none"> <li>▪ Understand the JavaScript and technical concepts behind Node JS</li> <li>▪ Structure a Node application in modules</li> <li>▪ Understand and use the Event Emitter</li> <li>▪ Build a Web Server in Node and understand how it really works</li> <li>▪ Use npm and manage node packages</li> <li>▪ Build a web application and API more easily using Express</li> <li>▪ Understand how to Connect to database in Node</li> </ul>				
<b>Pre-Requisites:</b> Basic Knowledge of JavaScript and OOPS				
Sr. No	Topics	Details	Weightage in %	Approx Lectures
1	<b>Introduction, Set up Development Environment, Other JavaScript Based Technologies</b>	<ul style="list-style-type: none"> <li>• Features and advantages of Node JS</li> <li>• Traditional Web Server Model</li> <li>• Node.js Process Model,</li> <li>• Asynchronous programming with Node.js</li> <li>• Types of applications that can be developed using Node.js</li> <li>• Install Node.js on Windows</li> <li>• working in REPL</li> <li>• Node JS Console</li> <li>• Creating a Node File with JavaScript</li> <li>• Accessing a Node.js File Through the Command Line Interface</li> <li>• Using Node.js in IDE</li> <li>• Node.js vs JavaScript</li> <li>• Node.js vs AJAX</li> <li>• Node.js vs JQuery</li> <li>• Node.js vs Angular JS</li> </ul>	<b>20</b>	<b>12</b>
2	<b>Node.js Basic, Node.js Modules, Node Package Manager (NPM)</b>	<ul style="list-style-type: none"> <li>• Primitive Types</li> <li>• Object Literal, Functions, Buffer, Access Global Scope</li> <li>• Module, Module Types: Core Modules, Local Modules, Third Party Modules, Module Exports.</li> <li>• Using Modules in a Node.js File</li> <li>• Using the Built in HTTP, URL, Query String Module</li> <li>• Creating a Custom Module</li> </ul>	<b>20</b>	<b>12</b>

**M.Sc. (IT & CA)**  
**Saurashtra University**  
**Effective from June - 2023**

		<ul style="list-style-type: none"> <li>• NPM, Installing Packages Locally</li> <li>• Adding dependency in package.json</li> <li>• Installing packages globally</li> <li>• Updating packages</li> </ul>		
<b>3</b>	<b>Creating Web Server, File System, Debugging Node.js Application</b>	<ul style="list-style-type: none"> <li>• Handling HTTP requests</li> <li>• Sending requests</li> <li>• Reading, Writing a File</li> <li>• Writing a file asynchronously</li> <li>• Opening a file</li> <li>• deleting a file</li> <li>• Other IO Operations: Append, Rename, Truncate</li> <li>• File System Module with URL Module Create, Read, Remove a Directory</li> </ul>	<b>20</b>	<b>12</b>
<b>4</b>	<b>Event, Database Connectivity</b>	<ul style="list-style-type: none"> <li>• EventEmitter class</li> <li>• Methods and Events of EvenEmitter Class</li> <li>• Returning event emitter</li> <li>• Extend EventEmitter Class</li> <li>• Passing Arguments and 'this' to listeners</li> <li>• Asynchronous and Synchronous call</li> <li>• Handle Events only Once, Error Events</li> <li>• Connection string for database connectivity,</li> <li>• Configuring, Working with insert, select command, Updating records, Deleting records, Drop tables, Ordered Result Set</li> </ul>	<b>20</b>	<b>12</b>
<b>5</b>	<b>Express and Node.js</b>	<ul style="list-style-type: none"> <li>• Introduction to Express Framework</li> <li>• Express Server Request-Response Routes</li> <li>• Route Parameters</li> <li>• Multiple Route Callback/Handler Functions</li> <li>• Methods of Response Object</li> <li>• Chaining Route Handlers</li> <li>• Send Static Files</li> <li>• Accept User Input</li> <li>• File Upload with Express</li> <li>• Manage Cookies</li> <li>• Send file as a response</li> <li>• Templates and Express.</li> </ul>	<b>20</b>	<b>12</b>
		<b>Total</b>	<b>100</b>	<b>60</b>

<b>References Books</b>
-------------------------

- |   |
|---|
| <ol style="list-style-type: none"><li>1. Dhruvi Shah, "Node.JS Guidebook", BPB Publications, 2018.</li><li>2. Sebastian Springer, Node.js: The Comprehensive Guide (Grayscale Indian Edition) Paperback - Shroff/Rheinwerk; First edition,2022</li><li>3. <a href="https://nodejs.org/en/docs/">https://nodejs.org/en/docs/</a></li></ol> |
|---|

**Course Outcome:**

After completion of the course students will be able:

- Understand Node JS and REPL terminal.
- Experiment with Node JS Modules and Node Package Manager.
- Develop applications to handle events in Node JS
- Make use of Web Server to manage database.
- Demonstrate Express Framework



**CS – 15: WEB DEVELOPMENT USING ANGULAR FRAMEWORK**

**Objectives:**

- Understand model view framework for building applications.
- Create modules for binding the application.
- Understand dependency injection for implementing services.
- Create and establish routes redirects and navigation.
- Validate forms for the submission of data.

**Pre-Requisites:** Basic Knowledge of HTML, JavaScript and TypeScript

Sr. No	Topics	Details	Weightage in %	Approx Lectures
1	<b>Introduction of Angular, Set up Development Environment, Basic of Angular</b>	<ul style="list-style-type: none"> <li>• Introduction to Angular</li> <li>• AngularJS vs Angular</li> <li>• MVC Framework</li> <li>• Component Based Model</li> <li>• Setting Up Angular</li> <li>• Installation of Node and NPM</li> <li>• Angular CLI</li> <li>• Creating and Running Project</li> <li>• Dependencies</li> <li>• App Component</li> <li>• Anatomy of Component</li> <li>• Creating Components</li> </ul>	<b>20</b>	<b>12</b>
2	<b>Data Binding in Angular</b>	<ul style="list-style-type: none"> <li>• Introduction to Data Binding</li> <li>• Types of Binding</li> <li>• Property Binding</li> <li>• Property Binding vs String Interpolation</li> <li>• Event Binding</li> <li>• Binding Data from Component</li> <li>• Async template Interpolation</li> <li>• Two-Way Binding</li> <li>• Forms Module and Two Way Binding</li> <li>• Understanding Directives</li> <li>• Looping with ngFor</li> </ul>	<b>20</b>	<b>12</b>

**M.Sc. (IT & CA)**  
**Saurashtra University**  
**Effective from June - 2023**

		<ul style="list-style-type: none"> <li>• Condition with ngif,</li> <li>• Passing inputs and variables to Components</li> <li>• ngModel for 2-way binding</li> <li>• ngOnInit</li> <li>• Styling with components</li> <li>• Creating multiple modules</li> </ul>		
<b>3</b>	<b>Dependency Injection and service in Angular</b>	<ul style="list-style-type: none"> <li>• Understanding Dependency Injection (DI)</li> <li>• Services</li> <li>• Creating a Service</li> <li>• Injecting the service into components</li> <li>• Understanding Dependency Hierarchical Injector</li> <li>• Injecting a Service into other services</li> <li>• Service Injection Context</li> <li>• Rest Calls with HttpClient</li> <li>• Building Angular Project</li> </ul>	<b>20</b>	<b>12</b>
<b>4</b>	<b>Routing &amp; Wrap Up</b>	<ul style="list-style-type: none"> <li>• Understanding the need of a Router</li> <li>• Setting Up and implementing Routes</li> <li>• Navigating to Router Links</li> <li>• Understanding Router Paths</li> <li>• Styling Active Router Links</li> <li>• Understanding Navigation Paths</li> <li>• Styling Router Links</li> <li>• Navigating Dynamically</li> <li>• Using Relative Paths</li> <li>• Passing Parameters to Routes and fetching route parameters</li> <li>• Fetching route parameters in a Reactive Way</li> <li>• Passing query parameters and fragments</li> <li>• Understanding Nested Routes</li> <li>• Redirecting and Wildcard routes</li> <li>• Wrap Up</li> </ul>	<b>20</b>	<b>12</b>
<b>5</b>	<b>Form Handling and Even Handling</b>	<ul style="list-style-type: none"> <li>• Introduction to Form Handling</li> <li>• Form Validation</li> <li>• ng-minlength</li> <li>• ngmaxlength</li> <li>• ng-pattern</li> <li>• ng-required</li> <li>• Submitting Forms</li> <li>• Event Handling with Forms.</li> </ul>	<b>20</b>	<b>12</b>
		<b>Total</b>	<b>100</b>	<b>60</b>

<b>References Books</b>
<ol style="list-style-type: none"><li>1. Angular 6 by Example: Get up and running with Angular by building modern real-world web apps, 3rd Edition, by Chandermani Arora.</li><li>2. Pro Angular 6, Apress, by Adam Freeman</li><li>3. Angular JS by Green, Orielly</li><li>4. Professional AngularJS (WROX), by Valeri Karpov</li></ol>



**Course Outcome:**

After completion of the course students will be able:

- Create Angular component using angular dependencies.
- Apply data binding objects for implementing modules.
- Create service and retrieve rest call data.
- Understand routes and their configuration in angular.
- Implement form handling with event driven apps.

**M.Sc. (IT & CA)**  
**Saurashtra University**  
**Effective from June - 2023**

<b>CS – 16: PRACTICAL - 1 (BASED ON CS-13)</b>	
<b>Topics</b>	<b>Marks</b>
FLUTTER APP DEVELOPMENT	<b>100</b>

<b>CS – 17: PRACTICAL - 2 (BASED ON CS-14 and (CS-15 OR CS-16))</b>	
<b>Topics</b>	<b>Marks</b>
<ul style="list-style-type: none"><li>• NODE JS</li><li>• WEB DEVELOPMENT USING ANGULAR FRAMEWORK OR</li><li>• APPLICATION DEVELOPMENT USING ASP.NET</li></ul>	<b>100</b>

**Note:**

- Practical examination may be arranged before or after theory exam.

<b>CS – 18: PROJECT DEVELOPMENT (In House)</b>	<b>Marks: 100</b>
Project must be developed in the computer laboratory of concern institute under the supervision of faculties of concern institute on any subject of current semester. <b><u>(At the time of Project-Viva examination student must show Project Report along with all the Workouts in workbook, implementation of project in SDLC, Documentation, Program codes and project in running mode)</u></b>	

**Note:**

- Project must be submitted before two weeks of commencement of theory exam.
- Project viva examination may be arranged before or after theory exam.
- During the project viva examination project must be run.

**M.Sc. (IT & CA)**  
**Saurashtra University**  
**Effective from June - 2023**  
**M.Sc. (IT & CA) (Semester – 4)**

<b>CS – 19: INDUSTRIAL PROJECT DEVELOPMENT</b>	<b>CREDIT - 30</b>	<b>Marks: 300</b>
<b>Project must be developed at industrial organization. <u>(At the time of Project-Viva examination student must show Project Report (In Hard Copy) along with all the Workouts in workbook, implementation of project in SDLC, Documentation, Program codes (Optional) and project in running mode).</u></b>		

**Guidelines:**

- (1) The project definition and finding industry should be started from the summer break after 2nd semester examinations.
- (2) Institute/College/Department has to make arrangement for the students for project development in various software development organizations in industry.
- (3) Project work must be developed at the industrial organization.
- (4) Internal guide from institute and external guide from Industry must be allocated to guide students. Internal guides (i.e. the faculty members) must devote the time to guide the students for the project.
- (5) Project plan along with the division of work amongst teammates would have been prepared and got approved within a maximum of 5 days of the start of the project.
- (6) Coding standards should be followed meticulously. At the minimum, the code should be self-documented, modular, and should use the meaningful naming convention.
- (7) It is advisable that object-oriented methodology is used with reusability of classes and code, etc.
- (8) The output reports must include MIS reports, if applicable.
- (9) The documentation should include a chapter on “Learning during Project Work”, i.e., “Experience of Journey during Project Duration”.
- (10) Data structure (database design) is mandatory. At least portions of code are mandatory. Student may be asked to write the code related to the project during examination.

<b>Sr. No</b>	<b>Evaluation Criteria</b>	<b>Marks</b>
1	EXPLANATION OF CODE	75
2	EXPLANATION OF ANALYSIS AND DESIGN	75
3	DOCUMENTATION	75
4	PRESENTATION	75
Total Marks		300