

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
1	UI Design	01
	1.1. Introduction to HTML5	01
	1.1.1 HTML5	01
	1.2. Key Features of HTML5	01
	1.2.1 Semantic Elements	01
	1.2.2 Multimedia Support	01
	1.2.3 Canvas API	02
	1.2.4 Improved Forms	02
	1.2.5 Local Storage	03
	1.2.6 Geolocation API	03
	1.2.7 Responsive Design	04
	1.3. Basic HTML5 Tags	04
	1.3.1. Structure of an HTML Document	04
	1.3.2. Heading Tags	05
	1.3.3. Paragraph and Text Formatting Tags	05
	1.3.4. Lists in HTML	06
	1.3.5. Links and Anchor Tags (<a>)	07
	1.3.6. Images ()	07
	1.3.7. Tables (<table>)	07
	1.3.8. Forms and Input Fields	08
	1.3.9. Line Breaks and Horizontal Rules	08
	1.4. HTML Forms and Input Elements	09
	1.4.1. Introduction to HTML Forms	09
	1.4.2. Common HTML Form Elements	09
	1.4.3. Buttons in Forms	12
	1.4.4. File Upload Field	12

CHAPTER NO	TITLE	PAGE NO
	1.4.5. Grouping Form Elements	13
	1.4.6. Form Attributes	13
	1.4.7. Form Validation	13
	1.4.8. Example: Complete Registration Form	14
	1.5. HTML Page Structure Elements	15
	1.5.1. Introduction	15
	1.5.2. Basic Structure of an HTML5 Page	15
	1.5.3. Explanation of Key Page Structure Elements	16
	1.5.4. Example: Full Web Page Layout	18
	1.5.5. Benefits of Using Semantic HTML Elements	19
	1.6. HTML Media Tags	20
	1.6.1. Introduction	20
	1.6.2. Images in HTML (Tag)	20
	1.6.3. Figures and Captions (<figure> and <figcaption>)	20
	1.6.4. Audio in HTML (<audio> Tag)	21
	1.6.5. Video in HTML (<video> Tag)	21
	1.6.6. Embedding YouTube and External Videos	22
	1.6.7. Responsive Media with CSS	22
	1.6.8. Full Example: Multimedia Webpage	22
	1.6.9. Summary	23
	1.7. Cascading Style Sheets (CSS)	24
	1.7.1. Introduction to CSS	24
	1.7.2. Types of CSS	24
	1.7.3. CSS Selectors	26
	1.7.4. CSS Properties and Values	27

CHAPTER NO	TITLE	PAGE NO
	1.7.5. CSS Box Model	28
	1.7.6. Responsive Design with CSS	29
	1.7.7. Example: CSS-Styled Web Page	29
	1.8. Introduction to Responsive Web Design	30
	1.8.1. Responsive Web Design (RWD)	30
	1.8.2. Core Principles of Responsive Web Design	30
	1.8.3. CSS Techniques for Responsive Design	31
	1.8.4. Responsive Design Frameworks	32
	1.8.5. Full Example: Responsive Web Page	33
	1.8.6. Summary	34
	1.9. Bootstrap – The Responsive Frontend Framework	34
	1.9.1. Introduction to Bootstrap	34
	1.9.2. Getting Started with Bootstrap	34
	1.9.3. Bootstrap Grid System	35
	1.9.4. Bootstrap Components	36
	1.9.5. Responsive Design with Bootstrap	37
	1.9.6. Full Example: Bootstrap Webpage	37
	1.10. Bootstrap Grid Basics	39
	1.10.1. Introduction to the Bootstrap Grid System	39
	1.10.2. Basic Structure of Bootstrap Grid	39
	1.10.3. Column Sizing in Bootstrap	40
	1.10.4. Bootstrap Grid Breakpoints	40
	1.10.5. Nested Grid (Grid inside a Grid)	41
	1.10.6. Aligning and Spacing Columns	41
	1.10.7. Full Example: Responsive Grid Layout	43
	1.11. Bootstrap Tables	43
	1.11.1. Introduction to Bootstrap Tables	43

CHAPTER NO	TITLE	PAGE NO
	1.11.2. Basic Bootstrap Table	44
	1.11.3. Striped Tables	44
	1.11.4. Bordered Tables	45
	1.11.5. Hoverable Tables	46
	1.11.6. Table Colors	47
	1.11.7. Responsive Tables	48
	1.11.8. Full Example: Stylish Bootstrap Table	49
	1.12. Images in Bootstrap	50
	1.12.1. Introduction to Bootstrap Images	50
	1.12.2. Responsive Images	50
	1.12.3. Image Shapes	51
	1.12.4. Image Alignment	51
	1.12.5. Image in a Grid Layout	51
	1.12.6. Image Backgrounds Using Bootstrap	52
	1.12.7. Example: Bootstrap Image Gallery	52
	1.12.8. Images in Web Development	53
	1.13. Buttons	56
	1.13.1. Introduction to Buttons	56
	1.13.2. Creating Buttons in HTML	57
	1.13.3. Creating Buttons with Links (<a> Tag)	57
	1.13.4. Styling Buttons with CSS	57
	1.13.5. Bootstrap Buttons	58
	1.13.6. Button Sizes	58
	1.13.7. Block-Level Button (Full Width)	58
	1.13.8. Button Groups	59
	1.13.9. Buttons with Icons	59
	1.13.10. Full Example: Button Showcase	59

CHAPTER NO	TITLE	PAGE NO
	1.14. Lists	61
	1.14.1. Introduction to Lists	61
	1.14.2. Types of Lists in HTML	61
	1.14.3. Ordered List ()	61
	1.14.4. Unordered List ()	62
	1.14.5. Description List (<dl>)	62
	1.14.6. Nested Lists (Lists Inside Lists)	63
	1.14.7. Styling Lists with CSS	63
	1.14.8. Bootstrap Lists	64
	1.14.9. Full Example: Interactive List with Bootstrap	65
	1.15. Drop-Down Menus	66
	1.15.1. Introduction to Drop-Down Menus	66
	1.15.2. Basic Drop-Down Using HTML (<select>)	66
	1.15.3. Multi-Select Drop-Down	67
	1.15.4. Grouping Options (<optgroup>)	67
	1.15.5. Customizing Drop-Downs with CSS	67
	1.15.6. Drop-Down Menu for Navigation (&)	68
	1.15.7. Bootstrap Drop-Downs	69
	1.15.8. JavaScript Drop-Down Menu	70
	1.15.9. Full Example: Interactive Drop-Down Menu	70
	1.16. Navigation	72
	1.16.1. Introduction to Navigation	72
	1.16.2. Basic Navigation Using HTML (<nav>)	72
	1.16.3. Responsive Navigation (Hamburger Menu)	73
	1.16.4. Bootstrap Navigation (. navbar)	74
	1.16.5. Navigation Tabs (.nav-tabs)	75
	1.16.6. Breadcrumb Navigation (. breadcrumb)	76

CHAPTER NO	TITLE	PAGE NO
	1.16.7. Full Example: Responsive Bootstrap Navbar with Tabs & Breadcrumbs	76
	1.16.8. Responsive Navbar with JavaScript	78
	1.16.9. Bootstrap Navbar (.navbar)	79
	1.16.10. Navbar with Dropdown Menu	80
	1.17. Forms and Input	81
	1.17.1. Basic HTML Form Structure	81
	1.17.2. Common Input Types in Forms	81
	1.17.3. Form with Different Input Types	82
	1.17.4. Placeholder vs Label in Forms	83
	1.17.5. Form Validation	83
	1.17.6. JavaScript Form Validation	83
	1.17.7. Bootstrap Styled Forms	84
	1.17.8. Example: Complete Contact Form with CSS	84
	1.17.9. Form Accessibility Best Practices	85
	1.17.10. Basic HTML Input Group	85
	1.17.11. Styling Input Groups with CSS	86
	1.17.12. Bootstrap Input Groups (.input-group)	87
	1.17.13. Common Use Cases for Input Groups	87
	1.17.14. Combining Multiple Input Elements	88
	1.17.15. File Input with Button (Bootstrap)	88
	1.17.16. Search Bar with Dropdown	88
	1.17.17. Form with Multiple Input Groups	88
	1.17.18. Accessibility Best Practices for Input Groups	89
	1.18. Modals	89
	1.18.1. Basic HTML & CSS Modal (Without JavaScript)	90
	1.18.2. JavaScript Modal (More Interactive)	91

CHAPTER NO	TITLE	PAGE NO
	1.18.3. Bootstrap Modal (.modal Component)	92
	1.18.4. Common Use Cases for Modals	93
	1.18.5. Login Modal Example (Bootstrap)	93
	1.18.6. Confirm Delete Modal	94
	1.18.7. Advanced: Modal with Video	94
2	JAVASCRIPT ES6	96
	2.1. Introduction to JavaScript ES6	96
	2.2. Variables in JavaScript ES6	96
	2.3. Operators in JavaScript ES6	96
	2.4. Control Structures in JavaScript	97
	2.5. Functions in JavaScript	97
	2.6. Scope in JavaScript	97
	2.7. Objects in JavaScript	97
	2.8. Arrays in JavaScript	98
	2.9. Date in JavaScript	98
	2.10. Math in JavaScript	98
	2.11. Regular Expressions (RegExp) in JavaScript	98
	2.12. HTML DOM in JavaScript	99
	2.13. Collections in JavaScript	99
	2.14. Event Handling in JavaScript	99
	2.15. JSON Parsing in JavaScript	99
3	Server Side JS Framework	27
	3.1. NODE.JS	100
	3.2. Why Node.js?	100
	3.3. Node.js Architecture	101
	3.4. Creating a Web Server with Node.js	102
	3.5. Handling Requests and Responses	102

CHAPTER NO	TITLE	PAGE NO
	3.6. Event Handling in Node.js	103
	3.7. GET and POST Methods	104
	3.7.1 What is GET and POST?	104
	3.7.2. GET Method	104
	3.7.3. POST Method	105
	3.7.4. Comparison: GET vs. POST	105
	3.7.5. When to Use GET and POST?	106
	3.7.6. Real-World Example: Login System	106
	3.8. File Upload	106
	3.8.1. What is File Upload?	106
	3.8.2. How Does File Upload Work?	106
	3.8.3. File Upload Using HTML and JavaScript	107
	3.8.4. File Upload with Node.js and Express (Backend Processing)	107
	3.8.5. File Upload with React.js (Frontend)	108
	3.8.6. Best Practices for Secure File Upload	109
	3.9. Connecting to a NoSQL Database using Node.js	109
	3.9.1. What is NoSQL?	109
	3.9.2. Why Use NoSQL with Node.js?	109
	3.9.3. Connecting to MongoDB using Node.js	110
	3.9.4. Fetch Data from MongoDB (Read Operation)	112
	3.9.5. Update and Delete Data in MongoDB	112
	3.9.6. Connecting to a Cloud NoSQL Database (MongoDB Atlas)	113
	3.9.7. Best Practices for Using NoSQL with Node.js	113
	3.10. Implementation of CRUD Operations in Node.js with MongoDB	113

CHAPTER NO	TITLE	PAGE NO
	3.10.1. What is CRUD?	113
	3.10.2. Setting Up the Project	113
	3.10.3. Setting Up MongoDB Connection	114
	3.10.4. Creating the User Model	114
	3.10.5. Implementing CRUD Operations	115
	3.10.6. Start the Server	117
	3.10.7. Best Practices	117
4	Typescript and Angular	118
	4.1. Typescript and Angular	118
	4.2. What is TypeScript?	118
	4.3. Definition of TypeScript	118
	4.4. Why Use TypeScript?	118
	4.5. TypeScript vs JavaScript: Key Differences	119
	4.6. Installing TypeScript	119
	4.7. Writing and Running TypeScript Code	120
	4.8. Features of TypeScript	120
	4.9. Variables in TypeScript	121
	4.10. Data Types in TypeScript	121
	4.11. Enum in TypeScript	121
	4.12. Arrays in TypeScript	122
	4.13. Tuples in TypeScript	122
	4.14. Functions in TypeScript	123
	4.15. Object-Oriented Programming (OOP) Concepts in TypeScript	123
	4.15.1. Classes and Objects	123
	4.15.2. Encapsulation (Data Hiding)	124
	4.15.3. Inheritance (Code Reusability)	125

CHAPTER NO	TITLE	PAGE NO
	4.15.4. Polymorphism (Method Overriding & Overloading)	126
	4.15.5. Abstraction (Hiding Implementation Details)	126
	4.15.6. Getters and Setters	128
	4.16. Interfaces in TypeScript	129
	4.16.1. Declaring an Interface	129
	4.16.2. Using Interfaces with Classes	130
	4.16.3. Optional and Readonly Properties	130
	4.16.4. Interface with Functions	131
	4.16.5. Extending Interfaces (Inheritance)	131
	4.16.6. Interface with Index Signature	132
	4.16.7. Implementing Multiple Interfaces	132
	4.16.8. Interface vs Type in TypeScript	133
	4.16.9. Real-World Example: API Response Interface	133
	4.17. Introduction to Angular	134
	4.18. Why is Angular Needed?	134
	4.19. Evolution of Angular	135
	4.20. Features of Angular	135
	4.21. Angular Architecture	135
	4.22. Setup and Configuration	136
	4.23. Components and Modules	136
	4.24. Templates	137
	4.25. Controllers	137
	4.26. Scope in Angular	137
	4.27. Directives	138
	4.28. Data Binding	138
5	Angular Js Framework	139

CHAPTER NO	TITLE	PAGE NO
	5.1. Pipes/Filters in AngularJS	139
	5.2. Filters in AngularJS	139
	5.3. Pipes in Angular (Angular 2+)	140
	5.4. Creating a Custom Pipe in Angular	141
	5.5. What is DOM (Document Object Model)?	141
	5.5.1. DOM Structure	142
	5.5.2. Types of DOM Nodes	142
	5.6. DOM Manipulation Using JavaScript	142
	5.7. DOM Events	143
	5.8. DOM in AngularJS	144
	5.9. Event	145
	5.10. Types of Events in JavaScript	145
	5.11. Event Handling in AngularJS	147
	5.12. Event Binding in AngularJS	148
	5.13. Routing	149
	5.14. Setting Up Routing in AngularJS	149
	5.15. Handling Route Parameters	151
	5.16. Redirecting Routes	151
	5.17. What Are Services in AngularJS	152
	5.17.1. Types of Services in AngularJS	152
	5.18. Creating a Custom Service	152
	5.19. When to Use Which Method?	154
	5.20. HTTP:	155
	5.21. Using \$http Service in AngularJS	155
	5.21.1. Common HTTP Methods in AngularJS	155
	5.22. Handling Errors	158
	5.22.1. Using \$http with Promises	158

CHAPTER NO	TITLE	PAGE NO
	5.23. Best Practices for Using \$http	158
	5.24. AJAX	159
	5.25. AngularJS Handles AJAX	159
	5.25.1. Using AJAX in AngularJS with \$http	159
	5.25.2. POST Request (Submitting Data)	160
	5.25.3. PUT Request (Updating Data)	161
	5.25.4. DELETE Request (Removing Data)	161
	5.26. Handling Errors in AJAX Requests	161
	5.26.1. AJAX Using \$http with Promises	162
	5.27. Template-Driven Forms	163
	5.27.1. Key Features of Template-Driven Forms	163
	5.27.2. Built-in Validation in Template-Driven Forms	165
	5.28. Handling Form Status in Template-Driven Forms	165
	5.29 Advantages of Template-Driven Forms	166
	5.30. Limitations of Template-Driven Forms	166
	5.31. What Are Reactive Forms?	166
	5.31.1. Key Features of Reactive Forms	166
	5.31.2. Dynamic Form Controls Using Form Builder	169
	5.31.3. Form Validation in Reactive Forms	170
	5.31.4. Advantages of Reactive Forms	170
	5.31.5. Limitations of Reactive Forms	170
	5.32. Comparison: Template-Driven vs. Reactive Forms	171
	5.33. Form Validation in Angular	171
	5.33.1. Types of Form Validation in Angular	171
	5.33.2. Built-in Form Validation in Angular	171
	5.33.3. Form Validation in Reactive Forms	172
	5.33.4. Built-in Validators in Angular	174

CHAPTER NO	TITLE	PAGE NO
	5.33.5. Custom Validators	174
	5.33.6. Asynchronous Validation	175
	5.34. Basic Animations in Angular	176
	5.34.1. Why Use Angular Animations?	176
	5.34.2. Setting Up Animations in Angular	176
	5.34.3. Angular Animation API	176
	5.34.4. Example: Fade In & Fade Out Animation	177
	5.34.5. More Examples of Angular Animations	178