Front end Development Lectureflow

Module 1 - Foundation	3

- How does the Internet Work
- DNS and how it works
- What is HTTP
- Browsers and how they work?
- What is Domain Name?
- What is hosting?
- Git and GITHUB Training

Module 2) Fundamentals of World Wide Web	3

- Careers in Web Technologies and Job Roles
- Difference between Web Designer and Web Developer
- What Is the Facebook Pixel
- How the Website Works
- Client and Server Scripting Languages
- Domains and Hosting
- Types of Websites (Static and Dynamic Websites)
- Web Standards and W3C recommendations
- Responsive Web Designing
- Protocol
- Basics of SEO
- SDLC

Module 1) Fundamentals of IT	3

- Careers in Web Technologies and Job Roles
- Difference between Web Designer and Web Developer
- What Is the Facebook Pixel
- How the Website Works
- Client and Server Scripting Languages
- Domains and Hosting
- Types of Websites (Static and Dynamic Websites)
- Web Standards and W3C recommendations
- Responsive Web Designing
- Protocol
- Basics of SEO
- Overview of SDLC
- Git and GITHUB Training

- Web development tools and environments are essential for building, testing, and deploying web applications. Here's an overview of some key tools and environments used in web development:
- Code Editors and Integrated Development Environments (IDEs) - Visual Studio, sublime, atom, jetbrains
- Version Control Systems - Git, Github, bitbucket
- Package Managers - NPM, Yarn, composer
- Task Runners and Build Tools - Grunt, Gulp, Webpack, Parcel
- Frameworks and Libraries - React, angular, vue.js, bootstrap, tailwind
- Development and Testing Servers - Node.js, Apache, Ngnix, Live Server
- Database management systems - MySQL, PostgreSQL, MongoDB, SQLite
- Front End Build Tools - SaSS, LESS, Babel
- Development Environments - XAMPP, MAMP
- Browser Developer Tools - Chrome DevTools, Firefox Developer Tools, Safari Developer Tools
- Collaboration and Communication Tools - trello, slack, Asana
- Continuous Integration and Deployment (CI/CD) Tool - Jenkins, Travis, CirceCI, Github Actionss
- Design and Prototyping Tools - Figma, Adobe XD, Sketch

Module 3) WD - HTML	**10**

- Student Intro , Career Center Login ,What is Internet, HTTP/HTTPS, WWW, Domain name and Top Domain name
- SEO, What is HTML, What is Text Editor, Web Browser, Downloading Text Editor , HTML Structure, First Program in HTML
- 1) HTML Introduction 2) HTML Getting Started 3) HTML Elements 4) HTML Attributes 5) HTML Basic Tags
- 1) HTML Doctypes 2) HTML Layout 3) HTML Head 4) HTML Meta 5) HTML Scripts
- Practical Examples: 1) Create any simple web page to display your name. 2) Importance of meta tag and Doctypes
- Tags and self Closing Tags, Basic Tag , Attribute and Events, Marquee Tag
- HTML - Meta Tags, HTML - Comments, HTML - Images, HTML - Tables, HTML - Lists, HTML - Text Links, HTML - Image Links
- HTML Headings HTML Paragraphs HTML Links HTML Text Formatting HTML Styles HTML Images
- HTML - Frames, HTML - Iframes, HTML - Blocks, HTML - Backgrounds, HTML - Colors, HTML - Fonts
- Anchor Tag, Img Tag, Image Mapping
- HTML - Fonts, HTML - Forms, HTML - Embed Multimedia ,HTML - Marquees, HTML - Header, HTML - Style Sheet, HTML - Javascript ,HTML - Layouts
- List Tag, Tables, Forms
- HTML - Tags Reference, HTML - Attributes Reference, HTML - Events Reference, HTML - Fonts Reference, HTML - ASCII Codes, ASCII Table, Lookup, HTML - Color Names, HTML - Entities, HTML - Fonts, Ref HTML - Events, Ref MIME Media Types, HTML - URL Encoding Language, ISO Codes HTML - Character Encodings, HTML - Deprecated Tags

- PRactical Examples: 1) Create simple Doc and display your name using different heading tag 2) Create link for open google. 3) Create document using all text formatting tags
- HTML online editor
- HTML Tables HTML Lists HTML Forms HTML Iframes
- Practical Examples: 1) Create simple table 2) Create time table for your school 3) Create table with colspanrowspan example 4) Create invoice using table 5) Create hotel menu. 6) Create index page for your book. 7) Create list with different categories.
- PRactical Examples: Create registration form with all fields and validation

Module 4) WD - CSS and CSS 3	**20**

- 1) CSS 2) In-line CSS Internal Style External Style Sheet @import Style Sheet 3) CSS Class CSS ID
- What is CSS How to Implement CSS Class and ID Width and Height Css Unit Box Model (Margin,padding,Border) and create basic template design
- Practical example : Create page with difference color text
- CSS Selectors , Pseudo Classes and Elements , Float and Clear and Alignment , Font Styling , Opacity and Visibility , Line Height
- 1) CSS Text 2)CSS Font 3) CSS Background 4) CSS Links 5) CSS Lists 6) CSS Display 7) CSS Visibility
- Creating Header of Website , Outline , Background , Counter increment , Counter reset ,Cursor , Overflow
- PRactical Example : Create layout for your project
- Position , Creating Submenu , Border Radius, Transform , Animation , Font Awesome Icons
- 1) CSS Layout Model 2) CSS Border 3) CSS Margin 4) CSS Padding 5) CSS Outline
- Font Family Through Google Font , import fontface rule ,FlexBox
- 1) CSS Float 2) CSS Align 3) CSS Position 4) CSS Element Size 5) CSS Layer
- Practical Example : Create image gallery
- 1) CSS Pseudo Class Selector 2) CSS Pseudo Element Selector
- CSS Properties 1) Background, 2) border 3) bottom 4) caption-side 5) clear 6) clip 7) color 8) content
- Practical Example: Create Menu with logo at left side and contact info at right side using clear effect
- 1) counter-increment 2) counter-reset 3) cursor 4) direction 5) display 6) empty-cells
- Practical Example: 1) Create submenu list using counter
- 1) float 2) font 3) height 4) left 5) letter-spacing 6) line [height, style, style-7) image, style-position, 8) style-type] 9) margin 10) outline 11) overflow 12) padding
- 1) page-break 2) position 3) quotes 4) right 5) table-layout 6) text 7) top 8) vertical-align 9) visibility 10) white-space 11) width 12) word-spacing 13) z-index
- Practical Example: wireframe layout for your template using div
- Responsive Design Principles, Media Query (For Responsive Website) , Creating a Responsive Website
- Validate a Website, Hosting a website with free domain name, Column , Clippath , Gradient Color , Filter, Border Image

Module 5) Website Designing - HTML5	**5**

- HTML5 Tags, HTML5 Input and Attribute
- Audio and Video, Semantic Element in HTML5
- Canvas, Svg
- Display Grid

Module 5) WD – Advance CSS/CSS Preprocessors	**10**

- Sass Variables
- strings · numbers
- colors · booleans
- lists
- nulls Sass Nested Rules
- Sass Importing Files
- Sass Mixins
- Sass @extend Directive
- ?Introduction to Sass: Explaining the basics of Sass, its syntax, and how it enhances traditional CSS.
- Variables and Variable Scope: Discussing the use of variables in Sass and how variable scope works.
- Mixins and @include: Explaining mixins and how they can be used for reusable styles, along with the @include directive.
- Nested Rules: Discussing nested rules in Sass and how they improve code organization and readability.
- Control Directives: Covering Sass control directives such as @if, @for, and @each, and how they can be used for conditional styles and loops.
- ?Functions: Explaining the use of functions in Sass for tasks like color manipulation, math operations, and more.
- Partials and @import: Discussing the use of partials to split Sass code into smaller, modular files, and how to import them using the @import directive.
- Extend/Inheritance: Explaining how to use the @extend directive in Sass for extending styles, and discussing its benefits and potential pitfalls.
- Operators: Covering Sass operators for arithmetic, logical operations, and string manipulation.
- Sass Maps: Introducing Sass maps and how they can be used for managing key-value pairs and organizing data.
- Mixins Libraries: Introducing popular Sass libraries like Bourbon or Compass and discussing their features and benefits
- ?Best Practices: Providing tips and best practices for writing clean, efficient, and maintainable Sass code.
- Less
- Variables
- Mixins
- Nesting
- Operations
- calc() exception
- Escaping
- Functions

- Namespaces and Accessors
- Maps
- Scope
- Comments
- Importing
- Introduction to Less CSS: Explaining the basics of Less CSS, its syntax, and how it extends traditional CSS
- Variables: Discussing the use of variables in Less CSS and how they contribute to code maintainability and reusability.
- Mixins and Functions: Explaining how mixins and functions work in Less CSS, and how they can be used to encapsulate styles and perform operations.
- Nesting: Discussing the nesting feature in Less CSS and its benefits for organizing styles and improving readability.
- Operations and Functions: Covering the various operations and functions available in Less CSS for arithmetic, color manipulation, and more.
- Imports and Partials: Explaining how imports and partials work in Less CSS to split stylesheets into smaller, more manageable files.
- Control Directives: Covering control directives like @if, @for, and @each in Less CSS and how they can be used to create dynamic stylesheets.
- Extend: Discussing the @extend directive in Less CSS and how it allows styles to be inherited from one selector to another.
- ?Nested Rules and Parent Selectors: Explaining nested rules and the use of the & selector in Less CSS for generating complex selectors.
- ?Mixins Libraries: Introducing popular Less CSS libraries like LessHat or Bootstrap Less and discussing their features and benefits.
- ?Best Practices: Providing tips and best practices for writing clean, efficient, and maintainable Less CSS code.
- ?SCSS (Sassy CSS) : - · Introduction to SCSS: Explaining what SCSS is, how it extends CSS, and its benefits. ·
- Variables and Mixins: Discussing the use of variables and mixins in SCSS to improve code reusability and maintainability.
- Nesting: Explaining how nesting works in SCSS and its advantages in organizing styles.
- Partials and Importing: Discussing the use of partials to split SCSS code into smaller files and the process of importing them into a main SCSS file.
- Control Directives: Covering the use of control directives like @if, @for, and @each to create more dynamic and efficient stylesheets.
- Functions: Explaining how functions in SCSS can be used to perform calculations, manipulate colors, and more.
- Inheritance and Extends: Discussing how inheritance can be achieved in SCSS using the @extend directive and its implications.
- Operators: Covering the various operators available in SCSS for performing arithmetic, logical, and other operations. ·
- Mixins Libraries: Introducing popular SCSS libraries like Bourbon or Compass and discussing their features and benefits.

- ?Best Practices: Providing tips and best practices for writing clean, efficient, and maintainable SCSS code.

Module 7) WD - JQuery Basic, Effects & Advanced	8

- jQuery Basic a) jQuery Introduction b) jQuery Getting Started c) jQuery Syntax d) jQuery Selectors e) jQuery Events
- What is JQuery , Downloading JQuery File , First Program in JQuery
- Practical Example: Change CSS
- JQuery Syntax , Query Selector, Hide , Slide , Fade Effect in JQuery
- JQuery Effects 1) jQuery Show/Hide 2) jQuery Fade 3) jQuery Slide 4) jQuery Animation 5) jQuery Stop 6) jQuery Chaining 7) jQuery Callback
- How to Apply CSS Using JQuery, How to Add Class and Remove Class in Jquery , JQuery Animation
- Practical Example: Create slider with animation
- Filter using JQuery , JQuery Slider Plugin , Validation Plugin
- JQuery Advanced 1) jQuery Traversing 2) jQuery Ancestors 3) jQuery Descendants 4) jQuery Siblings 5) jQuery Filtering 6) jQuery Load 7) jQuery No-Conflict
- Zoom Plugin, Now Make Your Existing Website Dynamic with Javascript and JQuery
- Ajax and ajax get post

Module 8) JavaScript Essentials And Advanced	25

- Basic JavaScript, Js comment, Js variables , Understanding var, let and Const, JS switch, if, else,JS loop , Js global variables, Js data types, Js operators, Js Functions
- Functions - Function Declaration in JS - Arrow Functions - Higher Order Functions - Map, Reduce and Filter
- Javascript Objects, Js object , Js Array , Js string, Js Date, Js Math, Js number, Js Boolean
- Javascript BOM ,Broswer Objects , Window object, History object, navigator object, Screen object
- Javascript DOM, Document object, getElementById, getElementByName, getElementByTagName, JS innerHTML property, JS innerTEXT property
- Javascript OOPS, JS class, JS object, JS prototype, JS constructor method, JS static method, JS encapsulation, JS inheritance, JS polymorphism, JS abstractions
- Javascript Exception Handling, JS exception handling , Javascript try-catch
- Javascript MISC, JS this keyword , JS Debugging , JS Hoisting , JS Strict Mode, JS promises, JS typeof , JS ternary operator, JS reload() method, JS setAttributes () method, JS setInterval() method, JS setTimeout() method.
- Javascript Events, Javascript Events, Javascript AddEventListener,(), jsOnclick event, jsdbclick event, JS onload event, JS onresize event.
- Array in JS, Creating Array, Array methods, The Spread & Rest operators, Destructuring
- JS Async, Callbacks, Promises, Async/Await
- ES6 Basics and Babel, New features in ES 6, Arrow functions, The . Operator, For/of , Map Objects, Set Objects, Promises, Functions Rest parameter, String.includes(),String.starts.With(), String.endWith(), Array.form(), Array.keys(), Array find(), Array findIndex(), javascript Modules

- Small Project using ES6
- Introduction to JavaScript Frameworks Overview of popular frameworks (React, Angular, Vue) Why use a framework?

Module-9) React - Components, State, Props	8

- •Different ways to installation – Create first project in react - Add React to a HTML Website - Create New React App - Hello World
- •React Js Introduction •Project Structure Overview
- •React Dom – Render HTML •SPA vs Traditional Web app
- JSX
- •Convert HTML to JSX
- Components
- Types of Components, Difference between Function Components and Class components
- Class Lifecycle
- Import - Export Components
- Working with Multiple components in same file, components nesting and composition
- Conditional Rendering
- •Ternary , if, if..else , switch case
- State and its importance
- •Commonly used events (onclick, onchange..)
- •Event Handlers
- •UseState – Hook introduction
- •Setstate in class components
- •Practical : accept value from user and display in h1 , create calculator app using state , hide – unhide div on button click
- •Prop and prop types
- •Updating Array in state
- •Updating object in state
- Basic understanding of Css – style

Module 10) Lists , Hooks , Localstorage , Api Project	6

- •Rendering List in React
- •Use of Map() function
- •Keys and importance in lists
- •useRef
- •React Refs
- •Uses of React Refs
- •useEffect practical
- •useContext
- •useReducer
- •useCallback
- •useMemo

- •Custom hook
- •Forms and Form handling
- •Form submission and validation
- •React Hook form
- •Project : Perform CRUD operations using array or map , render all records from array or map and display in table apply edit and delete operations – don't use any databases
- •JSON.parse and JSON.
- •Project : Perform CRUD operations using local storage database
- Project : Task manager application , pending task , completed task , select all , deselect all functionality

Module-11) React -Advance React- Styling , Routing	**5**

- •React bootstrap
- •Tailwind in React
- •Material UI in react js
- •Advantages of material ui in react js
- material UI components implementation
- •Buttons, Grid system , stack , typography , icons
- Customize style using material ui theme
- •Appbar
- •Layout components
- Drawer , grid layout
- •Form Components
- •Textfields , checkboxs , Form validation
- •Dialogs and modals
- •React Router – Browser – Router - Link , navlink
- •Route Parameters – useParams
- •Nested Routes
- Programmatic Navigation – history
- •Lazy loading (performance optimization)
- •Authentication – authorization
- •Redirecting unauthorized used to login page
- Using css transitions or animation library like React transition group.
- •Error Handling – 404 Page not found
- •Project : Project using API Or FakeJsonAPI
- Api testing in Postman

Module 12)React – JSON-server and Firebase Real Time Database	**8**

- •Installing Firebase SDK in react project
- •NoSql Database in firebase
- •Creating collection and documents
- •Authentication with firebase

- React router installation
- Use of firebase storage
- Project : Create app which contain registration , login and profile updation using firebase , user can post image and other user's can like dis-like post
- Project2: Web app using JSON-server

Module-13) React - Applying Redux	**8**

- State
- State storage problem
- Redux Basics
- Redux Principles
- Implementing Redux
- •Redux Core concepts – Actions , Reducers , Store React-Redux
- Middleware
- Counter App Demo
- Redux - Complexity of Managing state - Understand the Redux Flow - Setting up Reducer and store - Dispatching Actions - Passing and Retrieving Data with Action - Combining Multiple Reducers - Adding Middleware - Redux Dev tools
- •Redux-thunk middlerware
- Project : Perform CRUD operations using React-Redux