Fullstack UI/UX Syllabus

HTML Syllabus

Day 1:

1. Introduction of HTML
2. HTML Building blocks
    i. Tags
    ii. Attributes
    iii. Elements
3. HTML Formatting

Day 2:

4. Heading tags
    i. H1-h6 tags
5. HTML Paragraph
6. HTML Phrase Tags
7. HTML Anchor

Day 3:

8. HTML Image
9. HTML Table
10. HTML Lists
    i. Ordered List
    ii. Unordered List

Day 4:

11. HTML Form
    i. Attributes used for form tag
    ii. Form elements (input tag, textarea..)
12. HTML File Path
13. Head tags
    i. Link tag
    ii. Meta tag
    iii. Base tag
    iv. Script tag
    v. Style tag

   vi. Title tag

Day 5:

14. HTML Comments
15. HTML Entities, Symbols
16. HTML Iframes
    - i. Embed Youtube & Map

Day 6:

17. HTML Layout
    - i. Header tag
    - ii. Section tag
    - iii. Aside tag
    - iv. Article tag
    - v. Details , summary tag
    - vi. Footer tag
18. HTML Layout Techniques

Day 7 :

19. Computer code tags
20. Audio, Video
21. Project
22. Drag and Drop
23. Encoding URL
24. Handling of multiple file upload using multiple attribute
25. HTML Canvas

Day 8:

26. Geolocation API
27. Web Storage

CSS Syllabus

Day 9:

1. HTML with CSS
2. HTML Classes
3. HTML Id
4. CSS selectors

Day 10:

5. Creating Navigation Bars
6. CSS Image Sprites and Gradients

Day 11:

7. Applying animations, transitions to HTML elements
8. Creating 2D and 3D transformations
9. Media Query

Bootstrap Syllabus

Day 12:

1. Bootstrap introduction
2. Bootstrap features like fixed dropdown menu, carousel, text and image grids, custom thumbnails, bootstrap modal

Day 13:

3. Bootstrap Scroll Spy, Including jQuery in HTML pages

Javascript Syllabus

Day 14:

4. JavaScript introduction including DOM
5. JS Operators

Day 15:

6. JS Array
7. JS Loops, Statements, Switch Case

Day 16:

8. JS Functions
9. JS Functions with Parameters
10. JS Event handlers

Day 17 :

11. JS Event Listeners
12. Date () Math()

Day 18:

13. Validations
14. Project
15. Strict Mode
16. Cookie

Day 19:

17. Event bubling
18. Closures

Day 20:

19. Prototype
20. Promises

Day 21:

21. Async/Await

JQuery Syllabus

Day 22-23:

1. Jquery Basics
   i. Introduction
   ii. Download Jquery
   iii. Syntax
   iv. Selectors
   v. Methods
   vi. Events

Day 24-25:

2. Jquery Effects
   i. Show/hide
   ii. Fade
   iii. Slide
   iv. Animation
   v. Stop
   vi. Chaining
   vii. Callback
   viii. Insert Elements

Day 26-27:

3. Jquery Remove
   i. empty()
   ii. remove()
   iii. unwrap()
4. Jquery Dimensions
   i. innerWidth()
   ii. innerHeight()
5. Jquery Traversing
   i. ancestors
   ii. descendants
   iii. siblings
   iv. filtering
6. Creating a Countdown timer using jQuery timer API(Task)

Day 28:

7. jQuery widgets
    - i. accordion widgets
    - ii. tooltip widgets
    - iii. tabs widgets
    - iv. menu widgets

Day 29-30:

8. Using jQuery UI components e.g. Date picker into your HTML pages
9. jQuery Mobile
    - i. CSS Framework
    - ii. Mobile Events
    - iii. Mobile widgets

Day 31-32:

10. Jquery Ajax
    - i. ajax()
    - ii. load()
    - iii. get()
    - iv. post()
11. Project

React Syllabus

Day 33:

1) Environment Setup
    i. Installation of NodeJs and NPM
    ii. Installation of VSCode
    iii. Installation of React
    iv. Installation of React plugins in IDE
    v. Create React App
    vi. Create React Component
2) React Basics
    i. Building Blocks of Web ApplicationDevelopment
    ii. Single-page and Multi-page Applications
    iii. ECMAScript
    iv. Difference between ES5 andES6
    v. React DOM(Virtual DOM)
    vi. React Component
        a. Class Component
        b. Function Component
    vii. Component Lifecycle
    viii. State
    ix. Props
        a. Props with Class based Component
        b. Props with Function based Component
3) JSX
    i. What is JSX?
    ii. Why use JSX?
    iii. Rules of JSX implementation
    iv. Components with and without JSX(React.createElement)

Day 34:

4) React Developer Tools
5) Events
    i. Simple Events
    ii. Child events
    iii. Custom Events
    iv. Event Handling

Day 35-36:

6) Form
    i. Forms Validation

7) Router
    - i. Install React Router
    - ii. Create Components
    - iii. Add a router
    - iv. Navigation usingLinks
    - v. 404 page (Not foundPage)
    - vi. URLParameters
    - vii. Nested Routes
    - viii. Implementing styles usingNavLink
    - ix. Application ProgrammingInterface
    - x. Build a REST API usingjson-server
    - xi. API consumption in React application using Fetch method

Day 37:

8) List & Keys
    - i. Immutability
    - ii. Array immutability
    - iii. Object immutability

9) Closures

Day 38-40:

10) Lazy loading(Code splitting/Data fetching)
11) Lifting state up
12) Hooks
    - i. Basichooks
    - ii. useState()hook
    - iii. How to write useState() hook when state variable is an array of objects
    - iv. useEffect()hook
    - v. Fetch API data using useEffect()hook
    - vi. useContext()hook
    - vii. Rules to write React hooks
    - viii. Additional hooks
    - ix. Custom hooks

Day 41:

13) Introduction to Bootstrap
14) HOC
15) Context API

Day 42-46:

16) React Portal
17) React Redux
18) Fetch Data using GraphQL

- i. What isGraphQL?
- ii. Cons of RestAPI
- iii. Pros ofGraphQL
- iv. Frontend backend communication usingGraphQL
- v. Type system
- vi. GraphQLdatatypes
- vii. Modifiers
- viii. Schemas
- ix. GraphiQLtool
- x. Expressframework
- xi. NPM libraries to build server side ofGraphQL
- xii. Build a GraphQL API
- xiii. Apolloclient
- xiv. NPM libraries to build client side ofGraphQL
- xv. How to setup Apolloclient

19) Project

20) React Application Testing and Deployment

- i. Define Jest
- ii. Setup Testingenvironment
- iii. Add Snapshottesting
- iv. Integrate TestReducers
- v. Create TestComponents
- vi. Push Application on Git
- vii. Deploy App onNginx
- viii. Create Docker for ReactApplication

Node.js Syllabus

Day 47:

1. Node.js Introduction
   - i. What is Node.js?
   - ii. Why Node.js?
   - iii. Installing NodeJS
   - iv. Node in-built packages (buffer, fs, http, os, path, util,url)
   - v. Node.js Modules
     - a. Local Modules
     - b. Export Modules
     - c.

vi. Import your ownPackage
vii. Node Package Manager(NPM)
viii. Local and GlobalPackages
ix. Push code to GitHub

2. Node.js Console and Basics

Day 48:

3. Node.js Http Server
    i. JSON Data
    ii. Http Server and Client
4. Node.js File System
    i. Get Input fromUsers
    ii. Pass Multiple Arguments withYargs
    iii. File SystemModule
    iv. Operations associated with File SystemModule

Day 49:

5. Node.js Events
    i. Sending and receiving events with EventEmitters
6. ExpressJS and Web Application using ExpressJS
    i. ExpressFramework
    ii. Run a Web Server using ExpressFramework
    iii. Routes

Day 50:

7. Asynchronous Programming
    i. Call Stack
    ii. Callbacks, Callback Queue and EventLoop
    iii. CallbackAbstraction
    iv. CallbackChaining
    v. Promises
    vi. Promise Chaining
    vii. RequestPackage
    viii. Customizing HTTP Requests
    ix. Error handing with appropriate HTTPcodes
    x. Introduction to template engine(EJS)

Day 51:

8. Node.js with MongoDB
    i. Introduction to NoSQL Databases andMongoDB

ii. Installation of MongoDB onWindows

iii. Installation of Database GUIViewer

iv. InsertingDocuments

v. Querying, Updating and Deleting Documents

vi. Connect MongoDB and Node.jsApplication

vii. ExploringSendGrid

viii. Sending emails through Node.js application usingSendGrid

Day 52:

9. Node.js with SQL Server
10. REST APIs and GraphQL

i. RESTAPI
ii. REST API in Express
iii. Postman
iv. MongoDB DriverAPI
v. ExpressRouter
vi. Mongoose API
vii. GraphQL
viii. GraphQLPlayground

Day 53:

11. Building Node.js Applications using ES6

i. ES6 variables

ii. Functions withES6

iii. Import and ExportwithES6

iv. Async/Await

v. Introduction toBabel

vi. Rest API withES6

vii. Browsing HTTP Requests withFetch

viii. Processing QueryString

ix. Creating API usingES6

x. Transpilation

xi. Building DashboardAPI

xii. Creating dashboard UI withEJS

xiii. ES6 Aside: Default FunctionParameters

xiv. Data Validation andSanitization

Day 54:

12. User Authentication and Application
    i. Authentication
    ii. Types ofAuthentication
    iii. Session VsTokens
    iv. JSON WebTokens
    v. Bcrypt
    vi. Node-localstorage

Day 55:

13. Dynamic Client-Server Interaction using Socket.IO
    i. WebSockets
    ii. WebSockets
    iii. Socket.io
    iv. BroadcastingEvents
    v. Sharing YourLocation
    vi. EventAcknowledgements
    vii. Form and Button States
    viii. RenderingMessages
    ix. Working with Time and Timestamps for determining Location ofMessages
    x. Storing Users, Rendering User List, Tracking Users Joining andLeaving
    xi. Deploying the ChatApplication
    xii. Redis - Building API withRedis

Day 56:

14. Testing Node.js
    i. Writing Tests andAssertions
    ii. Testing AsynchronousCode
    iii. Testing an ExpressApplication
    iv. Setup andTeardown
    v. Testing withAuthentication
    vi. Advanced Assertions
    vii. MockingLibraries
    viii. Wrapping up UserTests
    ix. Setup Task TestSuite
    x. Testing with TaskData

Day 57:

15. Microservices

i. WhyMicroservices?
ii. What isMicroservices?
iii. WhyDocker?
iv. What isDocker?
v. Terminologies inDocker
vi. Child Processes
vii. Types of child process

Mongodb Syllabus

Day 58:

1. Introduction to NoSQL Database
   i. What in NoSQL?
   ii. Difference between NoSQL and RDBMS
   iii. Benefits of NoSQL
2. Introduction & Overview of MongoDB
   i. Objectives
   ii. Design Goals.
   iii. The Mongo Shell
   iv. JSON Introduction
   v. JSON Structure
3. MongoDB Installation
   i. Installing Tools
   ii. Overview of Blog Project.
   iii. Swig, Express
   iv. Node Packaged Modules (npm)

Day 59:

4. CRUD Operation in MongoDB
   i. CRUD (Creating, Reading & Updating Data) Mongo Shell
   ii. Query Operators
   iii. Update Operators and a Few Commands
5. Data Modeling
   i. Schema Design Pattern
   ii. Case Studies & Tradeoffs

Day 60:

6. Storage Classes

i. Automatic Storage Class
ii. Static Storage Class
iii. External Storage Class
iv. Register Storage Class

7. Indexing and Performance Considerations
   i. Performance Using Indexes,
   ii. Monitoring And Understanding Performance
   iii. Performance In Sharded Environments.

Day 61:

8. Aggregation
   i. Aggregation Framework Goals
   ii. The Use Of The Pipeline
   iii. Comparison With SQL Facilities.

Day 62:

9. MongoDB Replication
   i. Application Engineering Drivers
   ii. Impact Of Replication And Sharding On Design And Development.

Note: For completing the training in 62 working days we require to conduct UI/UX training for 4hrs every day.