GENERIC ELECTIVES (GE-6b): Internet Technologies: Web App Design and Development

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
GE6b: Internet Technologies: Web App Design and Development	**4**	**3**	**0**	**1**	Pass in Class XII	NIL

Learning Objectives

The course aims at:

- Develop understanding of Web Development Architecture.
- Using React components in Web applications
- Introduce REST APIs Design
- Understanding of Angular Architecture, data-binding and dependency injection
- Understand form validations and application of templates

Learning outcomes

On successful completion of the course students will be able to

- Develop interfaces for single page applications
- Develop a complete client side solutions using angular js
- Develop a RESTful web services.
- Apply form validations

SYLLABUS OF GE-6b

Unit 1

Introduction to React: Definition of React, React library, React Developer tools, Introduction to ES6, Declaring variables, Arrow Functions, Objects and Arrays, modules, Introduction to AJAX, Functions in AJAX Pure React: Page setup, virtual DOM, React Element, React DOM, Constructing Elements with Data, React Components, DOM Rendering, First React Application using Create React App, React with JSX, React Element as JSX Props, State and Component Tree: Property Validation, Validating Props with createClass, Default Props, ES6 Classes and stateless functional components, React state management, State within the component tree, state vs props, Forms in React

Unit 2

Rest APIs: JSON: Introduction, Syntax, Data Types, Objects, Schema. REST API:

Introduction, WRML, REST API Design, Identifier Design with URIs, Interaction Design with HTTP, Representation Design, Caching, Security.

Unit 3

Angular.js.: Introduction to Angular: Angular architecture; introduction to components, component interaction and styles; templates, interpolation and directives; forms, user input, form validations; data binding and pipes; retrieving data using HTTP; Angular modules

Essential/recommended readings

1. D. Brad, B. Dayley and C. Dayley, *Node. js, MongoDB and Angularjs Web Development: The definitive guide to using the MEAN stack to build web applications*, 2nd edition, Addison-Wesley, 2018.
2. D. Herron, *Node.js Web Development*, 5th edition, Packt Publishing, 2020.
3. A. Banks and E. Porcello, *Learning React: Functional Web Development with React and Redux*, 1st edition, O'Reilly, 2017.
4. M. Masse, *REST API – Design Rulebook*, 1st edition, O'Reilly, 2011.

Additional References

No additional references mentioned.

Suggested Practical List :

Practical exercises such as

1. Angular.js:
a. Build a simple Angular.js application that displays a list of items.
b. Create a form in Angular.js to add new items to the list.
c. Implement filtering and sorting functionality in Angular.js to manipulate the displayed list.
d. Integrate Angular.js with a RESTful API to fetch data and display it in the application.
e. Implement authentication and authorization using Angular.js routing and services.

2. React:
a. Create a basic React component that displays "Hello, World!" on the screen.
b. Build a React application that fetches data from a REST API and renders it in a list.
c. Implement form handling in React to create, update, and delete items from the list.
d. Create a search functionality using React to filter the displayed list based on user input.
e. Implement routing in React to navigate between different pages within the application.

3. REST API:
a. Build a simple REST API using a framework like Node.js and Express.
b. Create endpoints to perform CRUD operations (Create, Read, Update, Delete) on a specific resource (e.g., users, products).
c. Implement authentication and authorization mechanisms using JSON Web Tokens (JWT) to secure the API.
d. Develop endpoints that handle file uploads and downloads.