

UNIT – II**Backend Development with Node.js, Express, Angular & MongoDB**

1	a)	Identify the role of Node.js in backend development.	[L1][CO2]	[2M]
	b)	Identify the role of MongoDB in modern web applications	[L1][CO2]	[2M]
	c)	Define AJAX	[L1][CO2]	[2M]
	d)	Define Angular framework.	[L1][CO2]	[2M]
	e)	Differentiate between template-driven and reactive forms.	[L4][CO2]	[2M]
2	a)	Discuss the advantages of Node.js for scalable server-side application development.	[L2][CO2]	[5M]
	b)	Describe the structure of a basic Express.js application with a suitable example.	[L2][CO2]	[5M]
3	a)	Illustrate the implementation of routing in Express.js using different HTTP methods.	[L3][CO2]	[5M]
	b)	Differentiate between GET and POST requests with respect to Express.js routing.	[L4][CO2]	[5M]
4	a)	Explain Cross-Origin Resource Sharing (CORS) and justify its necessity in RESTful APIs	[L2][CO2]	[5M]
	b)	Compare the Node.js native HTTP module with the Express.js framework.	[L5][CO2]	[5M]
5	a)	Explain the concept of middleware in Express.js and analyze its role in the request–response cycle.	[L2][CO2]	[5M]
	b)	Describe the components of Mongoose: Schema, Model, and Document.	[L2][CO2]	[5M]
6	a)	Illustrate the creation of a database and collection using suitable examples.	[L3][CO2]	[5M]
	b)	Describe the steps involved in setting up a MongoDB database.	[L2][CO2]	[5M]
7	a)	Demonstrate insertion and retrieval of documents in MongoDB.	[L2][CO2]	[5M]
	b)	Explain the role of Mongoose in Node.js applications.	[L2][CO2]	[5M]
8	a)	Illustrate CRUD operations using Mongoose with suitable code snippets.	[L3][CO2]	[5M]
9	a)	Describe the structure and features of JSON with suitable examples.	[L6][CO2]	[5M]
	b)	Describe Angular components and their lifecycle.	[L2][CO2]	[5M]
10	a)	Differentiate between template-driven forms and reactive forms in Angular.	[L4][CO2]	[5M]
	b)	Analyze the end-to-end data flow from Angular frontend → Node.js server → MongoDB database.	[L4][CO2]	[5M]
11		Create a Contact Form using Express routes and MongoDB backend.	[L6][CO2]	[10M]

UNIT-III
React Framework & Full-Stack Integration

1	a)	Define React. List the features of ReactJS.	[L1][CO3]	[2M]
	b)	Identify the types of React components.	[L1][CO3]	[2M]
	c)	State the purpose of JSX.	[L1][CO3]	[2M]
	d)	State the role of Redux in React applications	[L1][CO3]	[2M]
	e)	State the purpose of Express.js in MERN stack.	[L1][CO3]	[2M]
2	a)	Develop a functional component using props and state.	[L6][CO3]	[5M]
	b)	Discuss the need for API communication in React apps.	[L2][CO3]	[5M]
3	a)	Describe the component lifecycle in React.	[L2][CO3]	[5M]
	b)	Construct a controlled form in React with example.	[L6][CO3]	[5M]
4	a)	Illustrate the useEffect() to fetch data from an API.	[L3][CO3]	[5M]
	b)	Differentiate the Redux store and reducer.	[L4][CO3]	[5M]
5	a)	Analyze controlled vs uncontrolled components.	[L4][CO3]	[5M]
	b)	Explain how Hooks simplify React development.	[L2][CO3]	[5M]
6		Build a React form with validation and submission feedback.	[L3][CO3]	[10M]
7	a)	Evaluate the advantages of Hooks over lifecycle methods.	[L5][CO3]	[5M]
	b)	Demonstrate API call using Axios in React.	[L2][CO3]	[5M]
8	a)	Develop routing using React Router.	[L6][CO3]	[5M]
	b)	Critique the use of Context API for complex state management.	[L5][CO3]	[5M]
9		Create a React → Express → MongoDB data flow diagram.	[L6][CO3]	[10M]
10	a)	Assess the importance of error handling in React apps.	[L5][CO3]	[5M]
	b)	Explain the steps involved in deploying React app using Netlify.	[L2][CO3]	[5M]
11		Design a React application with form validation and routing.	[L6][CO3]	[10M]

UNIT-IV
Advanced Frontend & Backend Integration

1	a)	Define RESTful API. List the principles of REST architecture.	[L1][CO5]	[2M]
	b)	Name common HTTP methods used in REST APIs.	[L1][CO5]	[2M]
	c)	What is role-based access control (RBAC)?	[L1][CO5]	[2M]
	d)	List types of files commonly uploaded in web applications.	[L1][CO6]	[2M]
	e)	Define pagination.	[L1][CO6]	[2M]
2	a)	Explain how HTTP methods map to CRUD operations.	[L2][CO5]	[5M]
	b)	Justify the use of JWT over server-side sessions.	[L5][CO5]	[5M]
3	a)	Explain the sequence: Client → Server → Database → UI update.	[L2][CO5]	[5M]
4	b)	Compare GET and POST methods with use cases.	[L4][CO5]	[5M]
5	a)	Justify the need for protected routes in frontend applications.	[L2][CO5]	[5M]
	b)	Describe how authorization differs from authentication.	[L2][CO5]	[5M]
6	a)	Explain how Multer handles multipart/form-data.	[L2][CO5]	[5M]
	b)	Differentiate between authentication token and authorization role.	[L4][CO5]	[5M]
7	a)	Evaluate Multer as a middleware for file handling.	[L5][CO6]	[5M]
	b)	Describe the difference between pagination and infinite scrolling.	[L2][CO6]	[5M]
8		Design a file upload module using Multer and MongoDB.	[L6][CO6]	[10M]
9	a)	Design a dynamic navigation bar for a full-stack app.	[L6][CO6]	[5M]
	b)	Develop a card-based UI integrated with paginated backend data.	[L6][CO6]	[5M]
10		Design a pagination and filtering API using MongoDB.	[L6][CO6]	[10M]

UNIT-V
Full-Stack Application Development & Deployment

1	a)	List the main layers in a React/Angular + Node + MongoDB application.	[L1][CO6]	[2M]
	b)	What are public routes and private routes?	[L1][CO6]	[2M]
	c)	Identify methods used to store and retrieve data.	[L1][CO6]	[2M]
	d)	Identify common validation rules.	[L1][CO6]	[2M]
	e)	Define media handling in web applications.	[L1][CO6]	[2M]
2	a)	Construct a standard folder structure for a MERN/MEAN project.	[L6][CO6]	[5M]
	b)	Classify layout routes for a common header and footer.	[L4][CO6]	[5M]
3	a)	Develop public and private routes using authentication.	[L6][CO6]	[5M]
	b)	Develop a routing strategy including nested, layout, and private routes.	[L6][CO6]	[5M]
4	a)	Describe different types of user feedback in forms.	[L2][CO6]	[5M]
	b)	Illustrate advanced validation rules in a form.	[L3][CO6]	[5M]
5	a)	Examine how poor error messaging affects user experience.	[L4][CO6]	[5M]
	b)	Differentiate between inline errors and global error messages.	[L4][CO6]	[5M]
6	a)	Build a form module with real-time error handling.	[L3][CO6]	[5M]
	b)	Differentiate between persistent and non-persistent state.	[L4][CO6]	[5M]
7	a)	Analyze localStorage vs sessionStorage.	[L4][CO6]	[5M]
	b)	Demonstrate serving images from backend to frontend.	[L2][CO6]	[5M]
8	a)	Compare frontend filtering and server-side filtering.	[L5][CO6]	[5M]
	b)	Illustrate MongoDB queries to filter data.	[L3][CO6]	[5M]
9	a)	Describe toast-based notification systems.	[L2][CO6]	[5M]
	b)	Build a notification and alert system for user feedback.	[L3][CO6]	[5M]
10		Create a toast and modal feedback workflow.	[L6][CO6]	[10M]
11	a)	Demonstrate deployment of a full-stack project.	[L2][CO6]	[5M]
	b)	Analyze common issues faced during project deployment.	[L4][CO6]	[5M]