

FULL STACK WEB DEVELOPMENT QUESTION BANK

P23CAT22

PART A – 2 Mark Questions with Answers | PART B – 16 Mark Questions

UNIT I – INTRODUCTION TO CSS AND JAVASCRIPT

PART A – 2 Mark Questions with Answers

Q.No	Question	Answer
1	What is the role of a Web Server?	A web server receives HTTP requests from clients and sends back responses (HTML, CSS, JS, images, etc.). Examples: Apache, Nginx, Node.js.
2	What is HTTP?	HTTP (HyperText Transfer Protocol) is the communication protocol used for transmitting data between web clients and servers over the internet.
3	What is the structure of an HTML Document?	An HTML document has a DOCTYPE declaration, <html> root element, <head> (metadata, title, links) and <body> (visible content) sections.
4	What are Basic Markup Tags in HTML?	Basic markup tags include <h1>–<h6> (headings), <p> (paragraph), <a> (link), (image), // (lists), <div>, .
5	What is CSS?	CSS (Cascading Style Sheets) is a stylesheet language used to control the visual presentation (layout, colors, fonts) of HTML elements.
6	What are CSS Selectors?	CSS selectors target HTML elements to apply styles. Types include element selector (p), class selector (.class), ID selector (#id), and attribute selector.
7	What is CSS Flexbox?	CSS Flexbox is a layout model that arranges elements in a flexible row or column, distributing space efficiently using properties like justify-content and align-items.
8	What is JavaScript?	JavaScript is a lightweight, interpreted scripting language used to add dynamic behavior, interactivity, and logic to web pages in the browser.
9	What are JavaScript Data Types?	JavaScript data types include: String, Number, Boolean, Null, Undefined, Symbol, BigInt (primitives), and Object (non-primitive).
10	What is an Event in JavaScript?	A JavaScript event is an action (e.g., click, keypress, mouseover) detected by the browser, which can trigger a function via event listeners.

11	What is AJAX?	AJAX (Asynchronous JavaScript and XML) allows web pages to send/receive data from a server asynchronously without reloading the whole page.
12	Differentiate GET and POST in AJAX.	GET retrieves data from the server (data in URL, visible, cacheable). POST sends data to the server (data in request body, secure, not cached).

PART B – 16 Mark Questions

Q.No	Question (16 Marks)
1	Explain the client-server architecture of the web. Discuss HTTP communication protocol and the structure of HTML documents with examples.
2	Explain CSS in detail: types of CSS, various selectors, box model, and CSS Flexbox layout with examples.
3	Describe JavaScript Data Types, Variables, Functions, and Event handling with suitable examples.
4	Explain AJAX in detail. Write code examples for both GET and POST methods using XMLHttpRequest or Fetch API.
5	Explain CSS Flexbox layout model with all its properties (flex-direction, justify-content, align-items, flex-wrap) and illustrate with examples.

UNIT II – SERVER SIDE PROGRAMMING WITH NODE JS

PART A – 2 Mark Questions with Answers

Q.No	Question	Answer
1	What is Node.js?	Node.js is an open-source, cross-platform JavaScript runtime built on Chrome's V8 engine, used for building server-side applications.
2	What is NPM?	NPM (Node Package Manager) is the default package manager for Node.js used to install, manage, and share JavaScript libraries and tools.
3	What is the http module in Node.js?	The http module in Node.js allows creation of HTTP servers and clients to handle web requests and responses without external dependencies.
4	What is Express.js?	Express.js is a minimal, fast, and flexible Node.js web application framework that provides routing, middleware, and server-side features.
5	What is Middleware in Express?	Middleware are functions in Express that have access to request (req), response (res), and next() to perform tasks like logging, authentication, and parsing.
6	What is Server-Side Rendering (SSR)?	SSR is the process where the server generates complete HTML pages and sends them to the client, improving SEO and initial page load speed.
7	What is a Templating Engine?	A templating engine (e.g., EJS, Pug, Handlebars) allows dynamic HTML generation on the server side by embedding JavaScript logic into HTML templates.
8	What are Static Files in Express?	Static files (HTML, CSS, JS, images) are served directly without processing. Express uses express.static() middleware to serve them.
9	What is async/await in Node.js?	async/await is a syntax for handling asynchronous operations. An async function returns a Promise; await pauses execution until the Promise resolves.
10	What is Fetching JSON from Express?	Express uses res.json() to send JSON responses. Clients can fetch this data using the Fetch API or AJAX to consume the API.
11	What is Routing in Express?	Routing defines how an Express app responds to HTTP requests (GET, POST, PUT, DELETE) at specific endpoints using app.get(), app.post(), etc.
12	What is the purpose of package.json?	package.json is the manifest file of a Node.js project containing metadata, scripts, dependencies, and version information.

PART B – 16 Mark Questions

Q.No	Question (16 Marks)
1	Explain Node.js architecture and the http module. Write a program to create a basic web server using Node.js.
2	Explain Express.js framework in detail: routing, middleware, request/response handling, and serving static files with examples.
3	Explain Server-Side Rendering with Templating Engines in Express.js. Compare EJS, Pug, and Handlebars with code examples.
4	Explain async/await in Node.js with examples. How does it improve asynchronous code readability compared to callbacks and promises?
5	Explain NPM in detail. Discuss package management, creating a package.json, installing dependencies, and using scripts.

UNIT III – ADVANCED NODE JS AND DATABASE

PART A – 2 Mark Questions with Answers

Q.No	Question	Answer
1	What is NoSQL?	NoSQL (Not Only SQL) refers to non-relational databases designed for large-scale, distributed data that does not fit into tabular relational models.
2	What is MongoDB?	MongoDB is a document-oriented NoSQL database that stores data in BSON (Binary JSON) format, offering flexibility and scalability.
3	What is MongoDB Shell?	MongoDB Shell (mongosh) is an interactive JavaScript interface for connecting to and querying a MongoDB database.
4	What is Request Body Parsing in Express?	Request body parsing extracts data sent in HTTP POST/PUT requests. Express uses <code>express.json()</code> and <code>express.urlencoded()</code> middleware for this.
5	What is Mongoose?	Mongoose is an ODM (Object Data Modeling) library for MongoDB in Node.js that provides schema-based modeling, validation, and query building.
6	How do you connect Node.js to MongoDB?	Using the Mongoose library: <code>mongoose.connect('mongodb://localhost/dbname')</code> establishes a connection between Node.js and MongoDB.
7	What are Cookies in web development?	Cookies are small pieces of data stored on the client's browser, sent with each HTTP request, used for session management, authentication, and tracking.
8	What is User Authentication?	User authentication is the process of verifying user identity, commonly using username/password with sessions, JWT tokens, or OAuth.
9	What is SQL in Node.js context?	SQL databases (like MySQL, PostgreSQL) can be used in Node.js via libraries like <code>mysql2</code> or <code>pg</code> to perform structured relational data operations.
10	What is a MongoDB Document?	A MongoDB document is a JSON-like record stored in a collection, analogous to a row in a relational table, with flexible key-value pairs.
11	What is Basic Querying in MongoDB?	Basic querying uses methods like <code>find()</code> , <code>findOne()</code> , <code>insertOne()</code> , <code>updateOne()</code> , <code>deleteOne()</code> on collections in MongoDB shell or via Mongoose.
12	What is a Session in Node.js?	A session stores user-specific data on the server side between HTTP requests, managed via <code>express-session</code> middleware with a session ID in cookies.

PART B – 16 Mark Questions

Q.No	Question (16 Marks)
1	Explain MongoDB system overview: data model, collections, documents, and basic querying using MongoDB shell with examples.
2	Explain how to connect MongoDB with Node.js using Mongoose. Discuss schema definition, CRUD operations, and validation with code examples.
3	Explain request body parsing in Express and how to add/retrieve data to/from MongoDB via a Node.js REST API.
4	Explain Cookie and Session management in Node.js. Write code to implement user login with session-based authentication.
5	Explain User Authentication in Node.js: implement registration, login, password hashing using bcrypt, and JWT-based token authentication.

UNIT IV – ADVANCED CLIENT SIDE PROGRAMMING (REACT JS)

PART A – 2 Mark Questions with Answers

Q.No	Question	Answer
1	What is React JS?	React is an open-source JavaScript library developed by Facebook for building fast, interactive user interfaces using a component-based architecture.
2	What is ReactDOM?	ReactDOM is the package that provides DOM-specific methods for React. ReactDOM.render() mounts a React component into a real DOM node.
3	What is JSX?	JSX (JavaScript XML) is a syntax extension for JavaScript that allows writing HTML-like code inside JavaScript, which is then transpiled by Babel.
4	What are Components in React?	Components are the building blocks of React UI. They are reusable, independent pieces of UI defined as functions or classes.
5	What are Props in React?	Props (properties) are read-only inputs passed from a parent component to a child component to customize its behavior or display.
6	What is State in React?	State is a built-in React object that holds dynamic data within a component. When state changes, the component re-renders automatically.
7	What is the Component Lifecycle in React?	React component lifecycle phases: Mounting (componentDidMount), Updating (componentDidUpdate), and Unmounting (componentWillUnmount).
8	What is the Fetch API in React?	The Fetch API is a browser built-in interface used to make HTTP requests and retrieve data asynchronously, commonly used in useEffect hooks in React.
9	What is Local Storage in JS?	localStorage is a Web Storage API that stores key-value pairs in the browser with no expiry, persisting even after the browser is closed.
10	What is Lifting State Up in React?	Lifting state up means moving shared state to the closest common parent component so multiple child components can access and update it.
11	What is Composition in React?	Composition is a pattern in React where components are combined by nesting them to build complex UIs, rather than using inheritance.
12	What is a React Hook?	Hooks are functions (like useState, useEffect) that let functional components use state and lifecycle features without writing class components.

PART B – 16 Mark Questions

Q.No	Question (16 Marks)
1	Explain React JS fundamentals: ReactDOM, JSX syntax, functional and class components, props, and state with code examples.
2	Explain React Component Lifecycle methods in class components. How are lifecycle features handled in functional components using hooks?
3	Explain the Fetch API in React with useEffect to retrieve data from a REST API and display it in components.
4	Explain state management in React: useState, Lifting State Up, and component composition with detailed examples.
5	Explain React Events and JS LocalStorage. Write a React application that stores and retrieves user preferences using localStorage.

UNIT V – APP IMPLEMENTATION IN CLOUD

PART A – 2 Mark Questions with Answers

Q.No	Question	Answer
1	What is Cloud Computing?	Cloud computing is the delivery of computing services (servers, storage, databases, networking, software) over the internet on demand.
2	Name major Cloud Providers.	Major cloud providers include Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and IBM Cloud.
3	What is a Virtual Private Cloud (VPC)?	A VPC is an isolated virtual network in a cloud environment where users can deploy resources with control over IP addressing, subnets, and routing.
4	What is Horizontal Scaling?	Horizontal scaling (scaling out) adds more machines/instances to handle increased load, improving capacity by distributing work.
5	What is Vertical Scaling?	Vertical scaling (scaling up) increases the resources (CPU, RAM, storage) of an existing machine to handle more load.
6	What is a Virtual Machine (VM)?	A VM is a software-based emulation of a physical computer that runs an OS and applications isolated from the host system.
7	What is Docker?	Docker is a platform for developing, packaging, and running applications in lightweight, portable containers that include all dependencies.
8	What is a Docker Container?	A Docker container is a lightweight, standalone executable package containing code, runtime, libraries, and settings needed to run an application.
9	What is Kubernetes?	Kubernetes (K8s) is an open-source container orchestration system that automates deployment, scaling, and management of containerized applications.
10	What is a Docker Image?	A Docker image is a read-only template used to create containers. It is built from a Dockerfile and stored in registries like Docker Hub.
11	What is Ethernet in cloud context?	In cloud infrastructure, Ethernet and virtual switches (vSwitches) connect VMs and containers within the cloud network fabric.
12	What is the difference between Docker and Kubernetes?	Docker creates and runs containers on a single host. Kubernetes orchestrates and manages multiple containers across a cluster of machines.

PART B – 16 Mark Questions

Q.No	Question (16 Marks)
1	Explain Cloud Computing: service models (IaaS, PaaS, SaaS), deployment models (Public, Private, Hybrid), and major cloud providers.
2	Explain Virtual Private Cloud (VPC): components (subnets, route tables, gateways), and discuss horizontal vs vertical scaling strategies.
3	Explain Docker in detail: architecture, Dockerfile, creating images, running containers, Docker Compose, and Docker networking.
4	Explain Kubernetes architecture: master node, worker nodes, pods, services, deployments, and ReplicaSets with examples.
5	Explain how to deploy a Full Stack application (React + Node.js + MongoDB) to the cloud using Docker and Kubernetes.