### LARAVEL **Basic Laravel Questions:**

1. What is Laravel, and why is it used?
2. Explain the MVC architecture in Laravel.
3. How do you install Laravel?
4. What is Artisan in Laravel?
5. What is Composer in Laravel?
6. How do you define routes in Laravel?
7. Explain Middleware in Laravel.
8. What is CSRF protection in Laravel, and how is it implemented?
9. What are Service Providers in Laravel?
10. Explain the concept of Eloquent ORM.

### **Routing and Middleware:**

1. How do you create a custom route in Laravel?
2. What is route grouping in Laravel?
3. Explain the difference between get, post, put, patch, delete, and options methods in routing.
4. How do you apply middleware to a route in Laravel?
5. What is route model binding in Laravel?
6. How do you create a custom middleware in Laravel?
7. Explain route caching and its benefits in Laravel.
8. How do you handle errors in routes?
9. What is route prefixing in Laravel?
10. How can you create RESTful routes in Laravel?

### **Controllers and Requests:**

1. What is a Controller in Laravel?
2. How do you create a Controller in Laravel?
3. What is the difference between resource and controller methods in route definition?
4. Explain the concept of Request Validation in Laravel.
5. How do you handle file uploads in Laravel?
6. What are Form Requests in Laravel?
7. How do you handle sessions in Laravel?
8. What is a policy in Laravel?
9. How do you use dependency injection in a Controller?
10. Explain RESTful controllers in Laravel.

### **Models and Eloquent ORM:**

1. What is Eloquent ORM?
2. How do you create a model in Laravel?
3. Explain the concept of relationships in Eloquent.
4. What are the different types of relationships in Eloquent?
5. How do you define a one-to-many relationship in Laravel?
6. Explain the concept of eager loading in Eloquent.
7. What is the difference between find and where in Eloquent?
8. How do you use hasManyThrough relationships?
9. What are accessors and mutators in Eloquent?
10. Explain the concept of soft deletes in Laravel.

### **Database and Migrations:**

1. What are migrations in Laravel?
2. How do you create a migration in Laravel?
3. What is the purpose of the php artisan migrate command?
4. How do you rollback a migration in Laravel?
5. Explain database seeding in Laravel.
6. What are database factories in Laravel?
7. How do you create a pivot table in Laravel?
8. How do you use database transactions in Laravel?
9. Explain the Schema facade in Laravel.
10. What is the difference between up and down methods in migrations?

### **Authentication and Security:**

1. How does authentication work in Laravel?
2. Explain the concept of guards in Laravel.
3. What are the different authentication drivers in Laravel?
4. How do you implement user authentication in Laravel?
5. What is password reset in Laravel, and how is it implemented?
6. Explain the concept of API authentication in Laravel.
7. How do you implement OAuth authentication in Laravel?
8. What is the purpose of the bcrypt function in Laravel?
9. How do you secure your Laravel application?
10. Explain the Auth facade in Laravel.

### **APIs and RESTful Services:**

1. How do you create a RESTful API in Laravel?
2. What is API Resource in Laravel?
3. How do you handle API versioning in Laravel?
4. Explain the concept of API throttling in Laravel.
5. How do you implement JSON Web Tokens (JWT) in Laravel?
6. What is Laravel Passport, and how do you use it?
7. How do you handle file uploads in a RESTful API?
8. What is the purpose of the apiResource method in routing?
9. How do you validate API requests in Laravel?
10. Explain the concept of rate limiting in Laravel.

### **Queues, Events, and Broadcasting:**

1. What are queues in Laravel, and why are they used?
2. How do you create a job in Laravel?
3. Explain the concept of event broadcasting in Laravel.
4. What is the purpose of the queue:work command in Laravel?
5. How do you implement real-time notifications in Laravel?
6. What is Laravel Echo, and how do you use it?
7. Explain the concept of jobs and workers in Laravel.
8. How do you handle failed jobs in Laravel?
9. What are events in Laravel, and how do you use them?
10. Explain the concept of listeners in Laravel.

### **Blade Templating and Frontend Integration:**

1. What is Blade templating engine in Laravel?
2. How do you extend layouts in Blade?
3. Explain the concept of components in Blade.
4. How do you include CSS and JS files in Blade?
5. What is the purpose of @yield and @section in Blade?
6. How do you handle loops and conditionals in Blade?
7. Explain the concept of Blade Directives.
8. How do you implement AJAX in Laravel?
9. What is the purpose of the @csrf directive in Blade?
10. How do you create a master layout in Blade?

### **Advanced Laravel Concepts:**

1. What are Service Containers in Laravel?
2. Explain the concept of dependency injection in Laravel.
3. How do you create a Service Provider in Laravel?
4. What is Laravel Horizon, and how is it used?
5. Explain the concept of task scheduling in Laravel.
6. What is the purpose of the config directory in Laravel?
7. How do you use environment variables in Laravel?
8. What is Laravel Mix, and how do you use it?
9. Explain the concept of localization in Laravel.
10. What is the purpose of the app directory in Laravel?

### ReactJs

### **Beginner Level React.js Questions**

1. What is React, and why is it used?
2. What are the key features of React.js?
3. What are components in React?
4. Explain the difference between functional and class components.
5. What is JSX, and why is it used in React?
6. What is the virtual DOM, and how does it work?
7. What is the difference between the real DOM and the virtual DOM?
8. How do you create a React component using ES6 syntax?
9. What are props in React?
10. How do you pass props between components in React?
11. What is the purpose of state in React components?
12. How do you manage state in a class component?
13. What is the difference between state and props?
14. How do you update the state in a React component?
15. What is the significance of the render() method in React?
16. What is the role of keys in React lists?
17. Explain how you can handle events in React.
18. What are controlled components in React?
19. What are uncontrolled components in React?
20. What is setState() in React, and how does it work?
21. How can you conditionally render components in React?
22. What are fragments in React?
23. Explain React lifecycle methods.
24. What is the purpose of componentDidMount()?
25. What is the purpose of componentDidUpdate()?
26. What is the purpose of componentWillUnmount()?
27. What is the difference between componentDidMount() and useEffect()?
28. What are synthetic events in React?
29. How do you handle forms in React?
30. How do you lift the state up in React?
31. What is the use of refs in React?
32. What are default props in React?
33. What is prop drilling in React, and how can you avoid it?
34. What are higher-order components (HOCs) in React?
35. Explain the use of inline styling in React.
36. How can you handle errors in React components?
37. What is the difference between useState and useReducer in React?
38. How do you pass functions as props in React?
39. Explain how React handles re-renders.
40. What is a functional update in useState()?
41. How do you force a component to re-render in React?
42. What is the purpose of React.memo()?
43. What is the role of PropTypes in React?
44. How can you optimize performance in React applications?
45. What is code splitting in React?

### **Intermediate Level React.js Questions**

1. What is React Router, and why is it used?
2. How do you implement routing in React using react-router-dom?
3. What is the difference between BrowserRouter and HashRouter?
4. What is lazy loading in React, and how do you implement it?
5. What is the useEffect hook, and how is it used?
6. What are custom hooks in React?
7. How do you create a custom hook in React?
8. Explain the difference between useEffect and useLayoutEffect.
9. What is the useContext hook, and how is it used?
10. How do you manage global state in React using the Context API?
11. What are React portals, and when would you use them?
12. How do you implement error boundaries in React?
13. What is reconciliation in React?
14. What is the purpose of shouldComponentUpdate() in React?
15. What is React's context, and how does it work?
16. How does React handle forms with multiple inputs?
17. What is the difference between useRef and createRef?
18. How do you create and use a ref in a functional component?
19. Explain how useMemo and useCallback work.
20. What is the difference between useMemo and useCallback?
21. How does React handle asynchronous operations in components?
22. What is the StrictMode component in React, and why is it used?
23. What is the purpose of defaultProps in React?
24. What are render props in React?
25. Explain how you can achieve composition in React.
26. What is the purpose of the children prop in React?
27. How do you handle side effects in React?
28. What are controlled vs uncontrolled forms in React?
29. What is the role of a fallback in React.lazy()?
30. How do you integrate third-party libraries into React?

### **Advanced Level React.js Questions**

1. How does the virtual DOM differ from the shadow DOM?
2. What is server-side rendering (SSR) in React, and why would you use it?
3. How do you implement SSR with React and Node.js?
4. What is Next.js, and how does it relate to React?
5. How do you manage state in a large-scale React application?
6. What is the role of redux-thunk in a React-Redux application?
7. What is Redux Saga, and how does it differ from Redux Thunk?
8. What is the useReducer hook, and how does it work?
9. How do you integrate Redux with React using hooks?
10. What is hydration in React, and why is it important for SSR?
11. What is a render prop pattern, and how does it work in React?
12. What are React hooks, and how do they differ from lifecycle methods?
13. Explain the concept of controlled and uncontrolled components.
14. What are some common performance optimization techniques in React?
15. How does React handle concurrent rendering?
16. What is React Fiber, and how does it improve React's performance?
17. What is React Suspense, and how does it work with lazy-loaded components?
18. How do you handle state management in React without Redux?
19. What is the Context API, and when would you use it over Redux?
20. What is a PureComponent in React, and how does it differ from a regular component?
21. How do you handle routing in server-side rendered React applications?
22. How can you create reusable components in React?
23. How does React implement reconciliation when updating the DOM?
24. What is the difference between React.StrictMode and React.Fragment?
25. How do you handle side effects with the useEffect hook in React?

MySqli Theory

### **Beginner Level MySQL Questions**

1. What is MySQL?
2. What are the key features of MySQL?
3. What is a database?
4. What is a table in MySQL?
5. What are the differences between SQL and MySQL?
6. How do you create a new database in MySQL?
7. How do you delete a database in MySQL?
8. What are MySQL data types?
9. How do you create a table in MySQL?
10. How do you drop a table in MySQL?
11. What is the CREATE command in MySQL?
12. What is the DROP command in MySQL?
13. How do you insert data into a table in MySQL?
14. How do you delete data from a table in MySQL?
15. How do you update data in MySQL?
16. How do you retrieve data from a table in MySQL?
17. What is the SELECT statement?
18. How do you use WHERE clause in MySQL?
19. What is the difference between WHERE and HAVING in MySQL?
20. What are JOINS in MySQL?
21. What are the different types of JOIN?
22. How do you use INNER JOIN in MySQL?
23. How do you use LEFT JOIN in MySQL?
24. How do you use RIGHT JOIN in MySQL?
25. How do you use FULL JOIN in MySQL?
26. What is GROUP BY in MySQL?
27. How do you use ORDER BY in MySQL?
28. How do you use LIMIT in MySQL?
29. How do you use DISTINCT in MySQL?
30. What are UNION and UNION ALL in MySQL?
31. What is the difference between UNION and UNION ALL?
32. How do you use LIKE in MySQL?
33. How do you use the IN clause in MySQL?
34. How do you use BETWEEN in MySQL?
35. What is the COUNT() function in MySQL?
36. How do you use the SUM() function in MySQL?
37. What is the AVG() function in MySQL?
38. How do you use MIN() and MAX() functions?
39. What is a PRIMARY KEY in MySQL?
40. What is a FOREIGN KEY in MySQL?
41. What are AUTO\_INCREMENT fields in MySQL?
42. How do you define relationships between tables in MySQL?
43. What is normalization in MySQL?
44. What are the different normal forms in database design?
45. What is the purpose of INDEX in MySQL?
46. How do you create an INDEX on a table in MySQL?
47. What is a unique index?
48. What are constraints in MySQL?
49. What is a DEFAULT constraint in MySQL?
50. How do you use CHECK constraint in MySQL?

### **Intermediate Level MySQL Questions**

1. How do you optimize a MySQL query?
2. What are subqueries in MySQL?
3. How do you write a subquery in MySQL?
4. What is a correlated subquery?
5. What is the difference between correlated and non-correlated subqueries?
6. What are stored procedures in MySQL?
7. How do you create and use stored procedures in MySQL?
8. What are triggers in MySQL?
9. How do you create a trigger in MySQL?
10. What are views in MySQL, and why would you use them?
11. How do you create a view in MySQL?
12. How do you update data in a view?
13. What are the limitations of MySQL views?
14. What is a transaction in MySQL?
15. How do you start a transaction in MySQL?
16. What are COMMIT and ROLLBACK in MySQL?
17. What are ACID properties in MySQL?
18. What are locks in MySQL?
19. How do you manage concurrency in MySQL?
20. What is deadlock, and how do you prevent it in MySQL?
21. What is a FULLTEXT index in MySQL?
22. How do you use a FULLTEXT index for searching?
23. What are MySQL storage engines?
24. What is the difference between InnoDB and MyISAM?
25. How do you switch storage engines in MySQL?
26. What are the advantages of using InnoDB over MyISAM?
27. How do you perform backups in MySQL?
28. What is the difference between logical and physical backups?
29. What is replication in MySQL?
30. How do you set up MySQL replication?
31. What are the different types of replication in MySQL?
32. How do you handle replication conflicts in MySQL?
33. How do you monitor MySQL performance?
34. What is query profiling in MySQL?
35. How do you use EXPLAIN to optimize queries in MySQL?
36. What is the difference between EXPLAIN and DESCRIBE?
37. How do you create a temporary table in MySQL?
38. What are the use cases for temporary tables?
39. What is the difference between a table alias and a column alias?
40. How do you perform batch processing in MySQL?
41. How do you calculate the difference between two dates in MySQL?
42. What are MySQL aggregate functions?
43. How do you convert data types in MySQL?
44. How do you use the COALESCE() function?
45. What is the IFNULL() function in MySQL?
46. What are common table expressions (CTEs) in MySQL?
47. How do you use the WITH clause in MySQL?
48. How do you handle errors in MySQL stored procedures?
49. What is the difference between CHAR and VARCHAR in MySQL?
50. How do you use DATE, TIME, and DATETIME in MySQL?

MySqli Practical

### **Beginner-Level Practical MySQL Questions**

1. Create a MySQL database named employee\_db.
2. Create a table employees with the following columns: id, name, age, department.
3. Insert five rows of data into the employees table.
4. Retrieve all records from the employees table.
5. Retrieve only the name and department from the employees table.
6. Update the age of an employee with id = 1 to 30.
7. Delete the employee with id = 2 from the employees table.
8. Add a new column salary to the employees table.
9. Retrieve all employees from the employees table where age > 25.
10. Find all employees who work in the HR department.
11. Retrieve the number of employees in the employees table.
12. Retrieve all employees, ordered by age in ascending order.
13. Retrieve the highest salary from the employees table.
14. Count how many employees are in the Sales department.
15. Create a departments table with columns id and name.
16. Write a query to retrieve employees along with their department names using a JOIN.
17. Write a query to retrieve employees whose name starts with 'A'.
18. Write a query to retrieve employees with age between 25 and 35.
19. Retrieve all distinct department names from the employees table.
20. Write a query to retrieve employees where the name contains the substring 'son'.

### **Intermediate-Level Practical MySQL Questions**

1. Write a query to find the second-highest salary in the employees table.
2. Write a query to retrieve the total salary paid to employees.
3. Write a query to find the average age of employees.
4. Create an index on the name column of the employees table.
5. Drop the index on the name column of the employees table.
6. Create a composite index on the name and department columns.
7. Write a query to retrieve employees grouped by department.
8. Use the HAVING clause to retrieve departments with more than 2 employees.
9. Write a query to find employees who don't have a department (NULL in the department field).
10. Write a query to delete all employees with a NULL salary.
11. Use a CASE statement to categorize employees based on age (e.g., 'Junior', 'Mid', 'Senior').
12. Retrieve the MIN() and MAX() salary from the employees table.
13. Write a query to update all employees in the IT department, giving them a 10% salary raise.
14. Retrieve all employees who were hired after a specific date.
15. Write a query to count employees by department using GROUP BY.
16. Use a subquery to find the employee with the highest salary.
17. Write a query to retrieve employees whose salary is higher than the department average.
18. Write a query to copy data from one table to another.
19. Write a query to create a temporary table temp\_employees with the same structure as employees.
20. Write a query to retrieve the last five records inserted into the employees table.
21. Write a query to perform a full outer join between two tables.
22. Write a query to retrieve employees along with the total salary per department.
23. Write a query to truncate the employees table.
24. Create a VIEW that shows only employees with salaries greater than 50,000.
25. Write a query to update a specific row using a WHERE clause in a JOIN query.
26. Write a query to delete rows from one table based on conditions from another table.
27. Write a query to count the number of distinct employee names in the employees table.
28. Write a query to retrieve the top 3 highest-paid employees from each department.
29. Write a query to calculate the number of years each employee has worked in the company.
30. Write a query to display the difference between the highest and lowest salary in each department.

### **Advanced-Level Practical MySQL Questions**

1. Write a stored procedure to insert a new employee into the employees table.
2. Create a trigger that automatically updates the submit\_date whenever a new row is inserted into the employees table.
3. Write a query to implement pagination on the employees table (show records 1-10).
4. Write a query to check if a specific value exists in the employees table.
5. Write a query to retrieve the last inserted id from the employees table.
6. Write a query to retrieve employees who were hired on the same day.
7. Create a stored procedure to calculate and return the total salary for a specific department.
8. Write a query to find gaps in sequential id values in the employees table.
9. Write a query to find duplicate rows in a table based on the name column.
10. Write a query to remove duplicate rows from a table.
11. Write a query to swap values between two rows in a table.
12. Create a MySQL function that takes an employee's ID as an argument and returns their department.
13. Write a query to schedule a recurring job using MySQL events.
14. Write a query to retrieve employees whose salary is in the top 10% of all salaries.
15. Create a partitioned table based on a date field in MySQL.
16. Write a query to retrieve records from multiple databases at once.
17. Write a query to calculate the difference between two dates in days.
18. Write a query to find employees who haven’t received a salary update in the last year.
19. Write a query to perform a recursive JOIN.
20. Write a query to disable foreign key checks.
21. Write a query to perform a transaction in MySQL, ensuring atomicity.
22. Write a query to ROLLBACK a transaction after a failure.
23. Write a query to check the current MySQL server version.
24. Write a query to calculate the total number of hours worked by all employees.
25. Write a query to back up a specific table.
26. Write a query to retrieve the most common value in a column.
27. Write a query to perform a case-insensitive search in the employees table.
28. Write a query to encrypt and decrypt data in a MySQL column.
29. Write a query to handle NULL values in arithmetic operations.
30. Write a query to retrieve only numeric data from a column that contains mixed data types.
31. Write a query to get the next available auto-increment value for a table.
32. Write a query to show all databases on the MySQL server.
33. Write a query to retrieve metadata about a MySQL table (column names, types).
34. Write a query to drop a column from a table.
35. Write a query to rename a column in a table.
36. Write a query to update all rows in a table based on values from another table.
37. Write a query to find all the users connected to the MySQL server.
38. Write a query to calculate the total number of employees hired in each month.
39. Write a query to implement a recursive common table expression (CTE).
40. Write a query to display the execution plan of a query.
41. Write a query to retrieve all employees who have been with the company for more than 5 years.
42. Write a query to copy all data from one database to another in MySQL.
43. Write a query to drop all tables in a database with a single command.
44. Write a query to add a foreign key constraint to a table.
45. Write a query to retrieve the average salary, grouped by department, but only for departments with more than 5 employees.
46. Write a query to change the character set and collation of a MySQL table.
47. Write a query to retrieve rows from a table where a specific column value occurs more than once.
48. Write a query to disable a unique index temporarily.
49. Write a query to remove all records older than one year from the employees table.
50. Write a query to calculate the running total of salaries for each department.