

Q1. React Basics (JSX, Components, Props)

User Story

As a **Product Page Developer**, I need a reusable ProductCard component that receives a product name, price, and discount as props so that the final discounted amount can be displayed on the ShopNow listing page.

Acceptance Criteria

- Functional component
 - Props: title, price, discount
 - Final price = price – discount
 - Parent component must pass props
 - JSX must display all fields clearly
-

Q2. React State + Controlled and Uncontrolled Components

User Story

As a **Login Experience Engineer**, I want the login form to capture the username through a controlled input while keeping the password field uncontrolled using useRef, so that the system can demonstrate both controlled and uncontrolled approaches.

Acceptance Criteria

- Username: controlled component
 - Password: uncontrolled using ref
 - On submit: log both values
 - Clear and structured JSX form
-

Q3. React Class Component, Lifecycle, PropTypes, Styling

User Story

As an **Account Status Developer**, I need a `UserStatus` class-based component that initially displays “Fetching user status...” and after two seconds shows “Active User,” ensuring that the component accepts a required numeric `userId` prop with styling applied.

Acceptance Criteria

- Class component
 - Lifecycle method updates message after 2 seconds
 - Uses `PropTypes` for validation
 - Includes basic CSS or inline styles
-

Q4. React Router + API Integration

User Story

As a **User Details Page Developer**, I need to configure routing so that `/users/:id` loads the `UserDetails` component which fetches user data from `http://localhost:4000/users/:id`, allowing individual user pages in the ShopNow admin panel.

Acceptance Criteria

- React Router setup
 - Parameter-based route
 - Fetching user data using the `id` param
 - Clean component structure
-

Q5. Reusability Using HOC or Render Props

User Story

As a **Frontend Performance Engineer**, I want a reusable utility (HOC or Render Props) that tracks the window width so that different components can display responsive UI behavior based on the same shared logic.

Acceptance Criteria

- Reusable logic for windowWidth
 - Implemented using Higher-Order Component or Render Props
 - Demonstrate usage in at least one component
-

Q6. Formik + Yup Validation

User Story

As an **Authentication Feature Developer**, I must build a login form using Formik and Yup where the email field is validated for proper format and the password field ensures a minimum of six characters to support basic input validation in ShopNow.

Acceptance Criteria

- Formik form
 - Yup validation schema
 - Submit handler
 - Clean UI layout
-

Q7. Node.js Core Modules

User Story

As a **Backend Logging Engineer**, I need a Node CLI script that writes “App started” into logs/app.log and starts a simple JSON HTTP server returning { "status": "running" }, using core modules so that the system is operational without external dependencies.

Acceptance Criteria

- Use fs to write log
 - Use path to create directory-safe file path
 - Create HTTP server
-

Q8. Asynchronous JavaScript (Callbacks → Promise → Async/Await)

User Story

As a **Modernization Engineer**, I must refactor the existing callback-based fetchData function into Promise-based and Async/Await-based versions so that the team follows updated asynchronous patterns.

Acceptance Criteria

- Correct Promise version
 - Correct Async/Await version
 - Correct output message
 - Maintain same functionality
-

Q9. Express Routing + Middleware + Validation

User Story

As a **Product API Developer**, I must build an Express server where /products returns a list of products, /products POST validates name and price using express-validator, and a global middleware logs request method and URL so that the API follows basic standards.

Acceptance Criteria

- GET and POST endpoints

- express-validator rules
 - Middleware logs method + URL
 - Return JSON responses
-

Q10. REST API + JWT Authentication

User Story

As an **Authentication System Developer**, I need to implement login and protected routes where a static user logs in through /login and receives a JWT token, which must be validated in middleware before allowing access to /dashboard.

Static User

- email: admin@test.com
- password: 12345

Acceptance Criteria

- JWT generation
 - Token validation middleware
 - Protected route works only with valid token
-

COMMON CONSOLIDATED RUBRIC (60 Marks)

Each question carries 6 marks distributed as:

Criteria	Marks
Correctness of implementation	2
Code structure, clarity, and JSX/Node patterns	2
Meeting all user story acceptance criteria	2
Total per question	6