

BÀI TẬP THUYẾT TRÌNH – BÀI KIỂM TRA SỐ 1

Các nhóm chọn nội dung và chuẩn bị dưới đây (Tối đa 3 nhóm / 1 nội dung)
Nhóm chọn trước được ưu tiên.

1. Hướng dẫn chung cho tất cả các nhóm

- **Thời lượng presentation:** 5–10 phút (trình bày lý thuyết + demo).
- **Báo cáo:** Slide (PDF/PowerPoint) & link repo (GitHub/GitLab).
- **Demo:** Chạy trực tiếp (local hoặc deploy trên Netlify/Vercel).
- **Tiêu chí chấm:**
 - Độ chính xác và sâu sắc của phần lý thuyết.
 - Chất lượng code (cấu trúc, readability).
 - Tính hoàn chỉnh & sáng tạo của demo.
 - Khả năng trả lời câu hỏi nhóm khác và giảng viên.

2. Nội dung:

Nhiệm vụ 1. JSX & Virtual DOM

1. Nội dung chủ đề giao cho sinh viên

- Giới thiệu JSX: cú pháp, cách biên dịch sang JavaScript thuần.
- Khái niệm Virtual DOM: hoạt động, so sánh với DOM thật.

2. Yêu cầu về nội dung lý thuyết trình bày

- Định nghĩa JSX, lợi ích và hạn chế.
- Cơ chế chuyển JSX → `React.createElement()` trong build.
- Nguyên lý hoạt động của Virtual DOM: diffing, reconciliation.
- Lợi ích của Virtual DOM về hiệu năng.

3. Gợi ý nội dung demo

- Một “JSX Playground”: cho phép gõ JSX và xem live preview.

- Mini app hiển thị một danh sách, thay đổi dữ liệu và in ra console quá trình diff giữa hai DOM.

4. Nội dung tổng kết

- JSX giúp code React giống HTML nhưng cần compile.
 - Virtual DOM giảm thao tác trực tiếp lên DOM thật, tăng hiệu năng re-render.
-

Nhiệm vụ 2. Component Paradigms: Functional vs Class & Props/State

1. Nội dung chủ đề giao cho sinh viên

- So sánh component dạng function và class.
- Cơ chế truyền data: props và quản lý nội bộ: state.

2. Yêu cầu về nội dung lý thuyết trình bày

- Khái niệm, cú pháp, lifecycle cơ bản của class vs function.
- Ưu/nhược điểm từng loại.
- Cách sử dụng props để truyền dữ liệu từ cha → con.
- Nguyên tắc bất biến (immutability) khi setState.

3. Gợi ý nội dung demo

- Hai phiên bản “Counter” một dùng class, một dùng function, so sánh code.
- Ứng dụng nhỏ “Profile Card”: cha truyền props name/avatar → con hiển thị.

4. Nội dung tổng kết

- Function components hiện đại, ngắn gọn, dễ test; class components vẫn dùng khi cần lifecycle phức tạp.
 - Props “read-only”, state quản lý nội bộ component.
-

Nhiệm vụ 3. Lifecycle Methods & useEffect

1. Nội dung chủ đề giao cho sinh viên

- Các lifecycle hooks trong class: `componentDidMount`, `componentDidUpdate`, `componentWillUnmount`...
- Tương đương với hook `useEffect` trong function component.

2. Yêu cầu về nội dung lý thuyết trình bày

- Trình tự gọi các phương thức lifecycle.
- Cú pháp và cơ chế hoạt động của `useEffect` với dependency array.
- Trường hợp cleanup (return function) trong `useEffect`.

3. Gợi ý nội dung demo

- Component đếm thời gian (timer):
 - Class: dùng `componentDidMount`/`componentWillUnmount`.
 - Function: dùng `useEffect` để `set/clear interval`.

4. Nội dung tổng kết

- `useEffect` là sự hợp nhất của nhiều lifecycle class, cho phép dễ quản lý side-effects.

Nhiệm vụ 4. `useState` & Các Hook Cơ bản

1. Nội dung chủ đề giao cho sinh viên

- Hook `useState`: cách khai báo, cập nhật state.
- Một số hook khác thường dùng: `useRef`, `useLayoutEffect`.

2. Yêu cầu về nội dung lý thuyết trình bày

- Cú pháp `const [value, setValue] = useState(initial)`.
- Nguyên tắc cập nhật state bất đồng bộ.
- Ứng dụng thực tiễn của `useRef` (giữ giá trị giữa các render) và `useLayoutEffect`.

3. Gợi ý nội dung demo

- Form nhỏ với một input text, demo update state realtime.
- Tạo một component focus input tự động bằng `useRef`.

4. Nội dung tổng kết

- useState đơn giản, phù hợp với hầu hết các nhu cầu state cơ bản.
 - Các hook khác hỗ trợ tối ưu render và thao tác với DOM.
-

Nhiệm vụ 5. Context API & Custom Hooks

1. Nội dung chủ đề giao cho sinh viên

- Context API: tạo, cung cấp, tiêu thụ context.
- Xây custom hook để tái sử dụng logic.

2. Yêu cầu về nội dung lý thuyết trình bày

- Cú pháp React.createContext(), Context.Provider và useContext.
- Khi nào nên dùng context, nhược điểm (re-render rộng).
- Tạo custom hook: nguyên tắc, cách tách logic.

3. Gợi ý nội dung demo

- Theme switcher (dark/light) dùng Context để truyền theme xuống nhiều component.
- Custom hook useWindowSize trả về kích thước viewport, sử dụng trong hai component khác nhau.

4. Nội dung tổng kết

- Context API giúp tránh “prop drilling”.
 - Custom hooks đóng gói logic, giữ code DRY.
-

Nhiệm vụ 6. React Router (v6+)

1. Nội dung chủ đề giao cho sinh viên

- Cài đặt, cấu hình React Router.
- Các component: BrowserRouter, Routes, Route, Link, useNavigate.

2. Yêu cầu về nội dung lý thuyết trình bày

- Khác biệt giữa v5 và v6.
- Nested routes, route params, redirect.
- Navigation programmatic với useNavigate.

3. Gợi ý nội dung demo

- Website 3 page: Home, About, Contact.
- Page Product detail: route /products/:id, hiển thị thông tin theo param.

4. Nội dung tổng kết

- React Router tổ chức điều hướng SPA, hỗ trợ nested và dynamic routes.
-

Nhiệm vụ 7. Form Handling & Validation

1. Nội dung chủ đề giao cho sinh viên

- Quản lý form trong React: controlled vs uncontrolled.
- Thư viện phổ biến: react-hook-form, Formik + Yup.

2. Yêu cầu về nội dung lý thuyết trình bày

- Khác biệt controlled/uncontrolled inputs.
- Cách setup react-hook-form: register, handleSubmit, formState.
- Validation schema với Yup; thông báo lỗi.

3. Gợi ý nội dung demo

- Form đăng ký user:
 - Các field: name, email, password.
 - Validate email đúng format, password tối thiểu 8 ký tự.
- Hiển thị inline error messages.

4. Nội dung tổng kết

- React-hook-form tối ưu performance cho form lớn.
 - Yup giúp tách riêng logic validate, dễ mở rộng.
-