

Lab 01: Revision of React

- Create a function component in separate file and link with App.js (A)
- Create a class component in separate file and link with App.js (B)
- Demonstrate the ReactJS props. (A)
- Demonstrate the Event Handling in ReactJS. (A)
- WAP in ReactJS to display the element if it has attribute called isDisplay to be true (using conditional rendering) (A)
- Demonstrate the use of map method in ReactJS to display array. (B)

Lab 02: Revision of React

- Display Faculties stored in array using ReactJS. (B)
- Display Students stored in array using ReactJS. (B)
- Display Products stored in array using ReactJS (C)
- Create a react application with following components.
 - create a component named “F” which print one state value named “name” from “App” Component.
 - create component named “E” which contains “F” component.
 - create component named “D” which contains “E” component.
 - create component named “C” which contains “D” component.
 - create component named “B” which contains “C” component with textbox and a button and when button is clicked set the state value named "name" from “App” component.
 - create component named “A” which contains “B” component.
 - "App" component should contains “A” component. (A)
- Create a react application with following components.
 - create a component named “F” which print one state value named “name” from “App” Component.
 - create component named “E” which contains “F” component.
 - create component named “D” which contains “E” component.
 - create component named “C” which contains “D” component.
 - create component named “B” which contains “C” component with button and when button is clicked set the state value named “name” from “App” component with the value of textbox from component “A”.
 - create component named “A” which contains “B” component and a textbox.
 - "App" component should contains “A” component. (B)
- Implement static login implementation with property drilling. (C)

Lab 03: Practicing Typography and spacing classes in tailwind

- Create a centered blog title with a subtitle below it. (A)
- Make a card with a heading, paragraph, and button — make them visually spaced properly. (A)
- Create a pricing card with a price highlighted using larger text. (A)
- Show a quote in italic with comfortable reading line-height. (B)
- Create simple nav links with equal spacing. (B)
- Display a user's name, role, and description with proper vertical spacing. (B)
- Create a page that displays all heading styles (H1–H6). (A)
- Create a Newsletter signup box with proper spacing. (B)
- Place two long text paragraphs side by side with readable spacing. (B)
- Show a product title, short description, features list with proper spacing. (C)

Lab 04: Practicing sizing classes in tailwind

- Create a 3-image gallery where images become full-width on small screens. (A)
- Create a product card with a fixed size box containing an image + text. (A)
- Create a page layout with a left sidebar and right content section. (B)
- Create 3 square boxes of different sizes. (A)
- Create a banner that fills the entire screen height with centered text. (A)
- Create a responsive text box that has a min width and max width. (B)
- Create 4 cards that increase width on medium screens. (C)
- Create a list inside a scrollable box. (C)
- Create a profile image placeholder. (B)
- Create a responsive video container that maintains 16:9 ratio. (B).

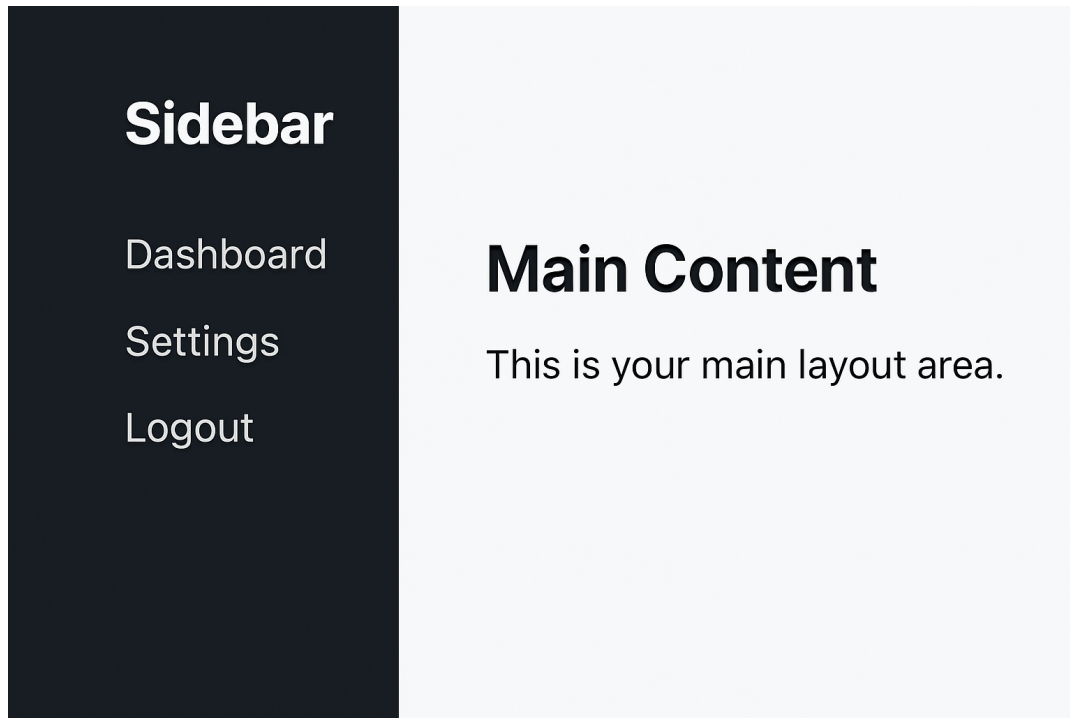
Lab 05: Practicing background, lines, effects and filters in tailwind.

- Create a hero banner with a left-to-right gradient background. (A)
- Create a card with a background image and white text overlay. (A)
- Create a glass-effect card on top of a colorful background. (B)
- Create an image that becomes blurred when hovered. (A)
- Create three boxes with different border styles and spacing. (B)
- Create 3 cards each using a different shadow depth. (B)
- Create a section with a background image and text readable using blend modes. (C)
- Create 3 images, each with a different filter, and remove the filter on hover. (B)
- Create a div with a pulsing animated gradient. (C)
- Create a layout showing horizontal and vertical separator lines. (B)

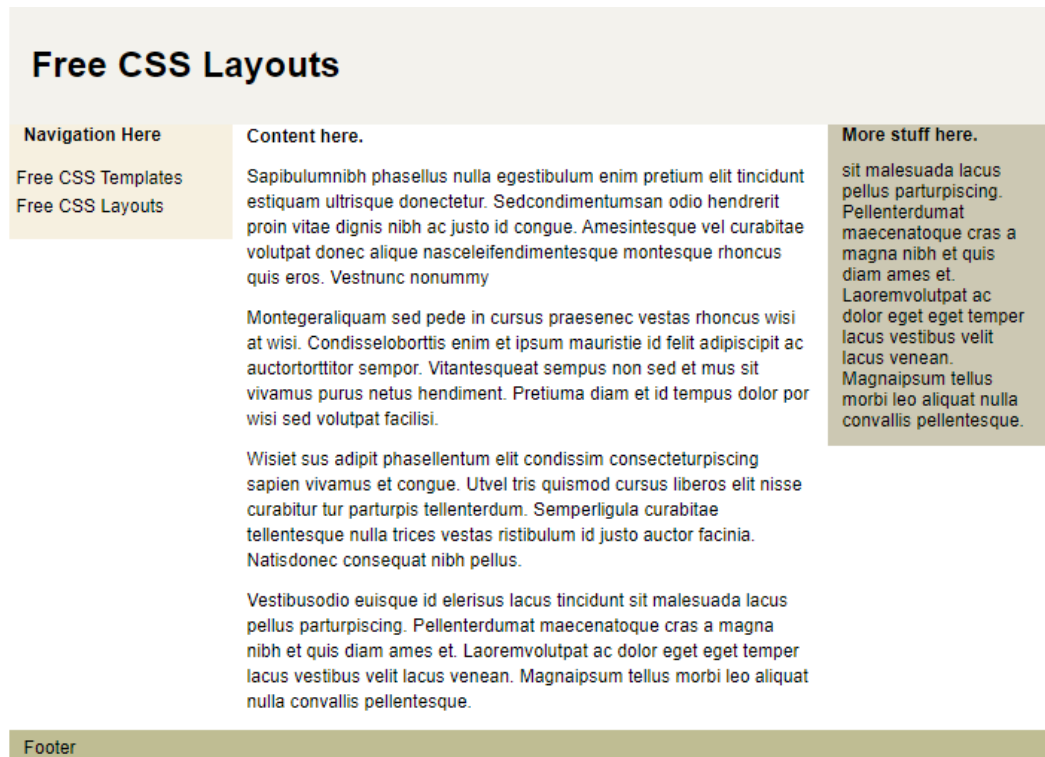
- Create a card that increases its ring size when hovered. (B)
- Make a card that lifts upward when hovered with shadow increase. (B)

Lab 06: Implementing layouts in tailwind.

- Implement below layout (A)



- Implement below layout using tailwind (A)



- Implement below layout using tailwind (B)

Free CSS Layouts

Content here.

Sapibulumnibh phasellus nulla egestibulum enim pretium elit tincidunt estiquam ultrisque donectetur. Sedcondimentumsan odio hendrerit proin vitae dignis nibh ac justo id congue. Amesintesque vel curabitaе volutpat donec aliquе nasceleifendimentesque montesque rhoncus quis eros. Vestnunc nonummy

Monteeraliquam sed pede in cursus praesenec vestas rhoncus wisi at wisi. Condisseloborttis enim et ipsum mauristie id felit adipiscipit ac auctortortitor sempor. Vitantesqueat sempus non sed et mus sit vivamus purus netus hendiment. Pretiuma diam et id tempus dolor por wisi sed volutpat facilisi.

Wisiet sus adipit phasellentum elit condissim conseceturpiscing sapien vivamus et congue. Utvel tris quismod cursus liberos elit nisse curabitur tur parturpis tellenterdum. Semperligula curabitaе tellentesque nulla trices vestas ristibulum id justo auctor facinia. Natisdonec consequat nibh pellus.

Vestibusodio eisque id elerisus lacus tincidunt sit malesuada lacus pellus parturpiscing. Pellenterdumat maecenatoque cras a magna nibh et quis diam ames et. Laoremvolutpat ac dolor eget eget temper lacus vestibus velit lacus venean. Magnaipsum tellus morbi leo aliquat nulla convallis pellentesque.

More stuff here.

sit malesuada lacus pellus parturpiscing. Pellenterdumat maecenatoque cras a magna nibh et quis diam ames et. Laoremvolutpat ac dolor eget eget temper lacus vestibus velit lacus venean. Magnaipsum tellus morbi leo aliquat nulla convallis pellentesque.

Footer

Navigation Here

Free CSS Templates

Free CSS Layouts

- Implement below layout using tailwind (C)

site title

fixed 1-column #2 layout (basic)

leftColumn

[Link Item](#)

[Link Item](#)

[Link Item](#)

Augur et fulgente decorus arcu Phoebus acceptusque novem Camenis, qui salutari levat arte fessos corporis artus.

centerColumn

Augur et fulgente decorus arcu Phoebus acceptusque novem Camenis, qui salutari levat arte fessos corporis artus, si Palatinas videt aequos aras remque Romanam Latiumque felix alterum in lustrum meliusque semper prorogat aevom, quaeque Aventinum tenet Algidumque, quindecim Diana preces virorum curat et votis puerorum amicas adplicat auris.

Header 2

[Link Item](#)

- List Item
- List Item
- List Item
 - List Item
 - List Item

blockquote

Augur et fulgente decorus arcu Phoebus acceptusque novem Camenis, qui salutari levat arte fessos corporis artus.

[XHTML](#) | [CSS](#) | [WCAG](#) | [The CSS Tinderbox](#)

- Implement below layout using tailwind (C)

[Home](#) [About](#) [Gallery](#) [Contact](#)

site title

fixed 3-column over 4-column layout

centerColumn

sub heading

Augur et fulgente decorus arcu Phoebus acceptusque novem Camenis, qui salutari levat arte fessos corporis artus, si Palatinas videt aequos aras remque Romanam Latiumque felix alterum in lustrum meliusque semper prorogat aevom, quaeque Aventinum tenet Algidumque, quindecim Diana preces virorum curat et votis puerorum amicas adplicat auris.

cell_1

Quindecim Diana preces virorum curat et votis puerorum amicas adplicat auris.

[Link](#)

Date

Puerorum amicas adplicat auris.

[Link](#)

Date

Puerorum amicas adplicat auris.

cell_2

Augur et fulgente decorus arcu Phoebus acceptusque novem Camenis, qui salutari levat arte fessos corporis artus, si Palatinas videt aequos aras remque.

Romanam Latiumque felix alterum in lustrum meliusque semper prorogat aevom, quaeque Aventinum tenet Algidumque, quindecim Diana preces virorum curat et votis puerorum amicas adplicat auris.

cell_3

Augur et fulgente decorus arcu Phoebus acceptusque novem Camenis, qui salutari levat arte fessos corporis artus, si Palatinas videt aequos aras remque.

Romanam Latiumque felix alterum in lustrum meliusque semper prorogat aevom, quaeque Aventinum tenet Algidumque, quindecim Diana preces virorum curat et votis puerorum amicas adplicat auris.

box_1

Quindecim Diana preces virorum curat et votis puerorum amicas adplicat auris.

[Link...](#)

box_2

Quindecim Diana preces virorum curat et votis puerorum amicas adplicat auris.

[Link...](#)

box_3

Quindecim Diana preces virorum curat et votis puerorum amicas adplicat auris.

[Link...](#)

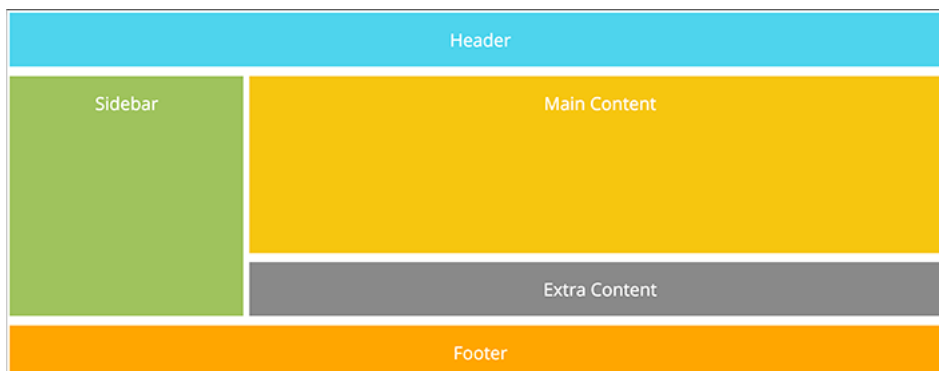
box_4

Quindecim Diana preces virorum curat et votis puerorum amicas adplicat auris.

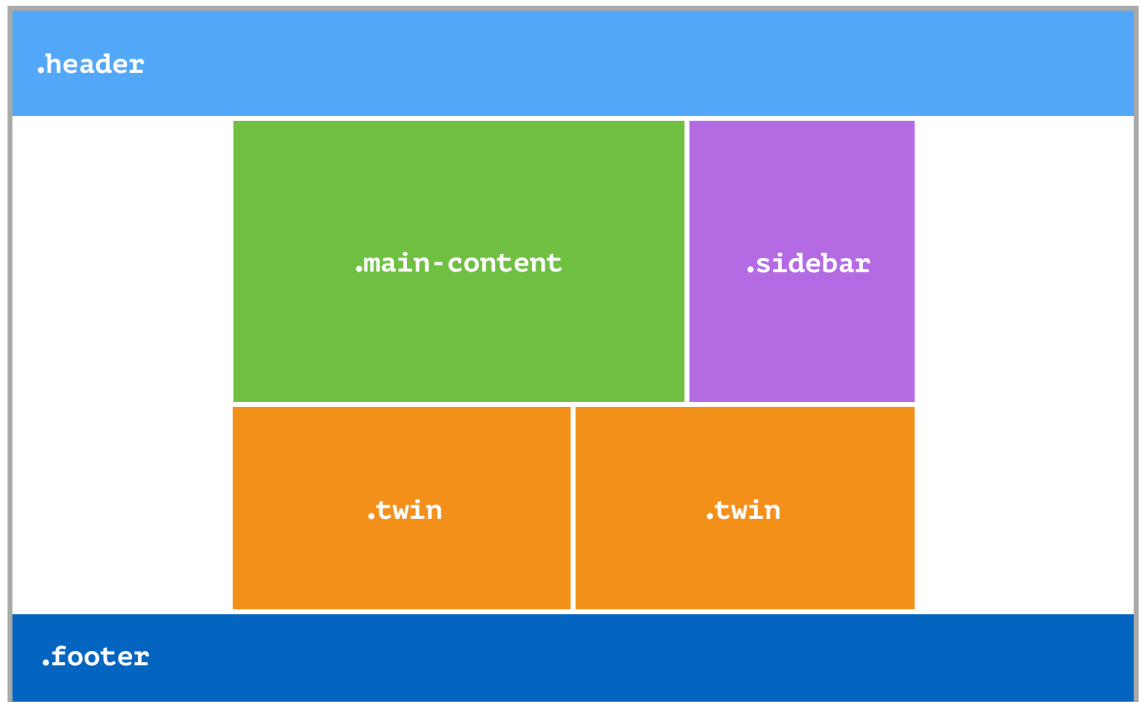
[Link...](#)

Copyright, © 2005-2006 [Company Name](#) | [The CSS Tinderbox](#)

- Implement below layout using tailwind (A)



- Implement below layout using tailwind (B)



Lab 07: practicing transactions and animation classes in tailwind.

- Create a button that changes color and slightly scales up on hover. (A)
- Make a hidden text fade in when card is hovered. (A)
- Create an image that zooms in slowly on hover. (A)
- Build a sidebar that slides in/out using a button toggle. (B)
- Create a notification badge with a pulsing glow. (B)
- Build a CSS-only spinner using Tailwind. (C)
- Put a download arrow icon that bounces repeatedly. (C)
- Simulate a typing effect with blinking cursor. (C)
- Create a section that fades in when page loads. (B)
- Create a button with a ripple animation when clicked. (C)

Lab 08: practicing table, svg, transforms and interactivity in tailwind.

- Create a simple table with borders and striped rows. (A)
- Create a responsive table which scroll on mobile view. (A)
- Display SVG icon with tailwind colors. (A)
- Animated SVG Hover Color Change. (B)
- Transform Scale on Hover. (B)
- Rotate element 45° on Hover. (B)
- Implement button with press effect. (B)

- Implement card with 3d hover tilt effect. (C)
- Design Interactive Checkbox Toggle State. (C)
- Design Interactive Menu (show/hide on hover). (C)

Lab 09: installing and setting up MUI in React project.

- Create a New React Project. (A)
- Install MUI Core. (A)
- Install MUI Icons. (A)
- Install Additional MUI Modules. (A)
- Write a detailed note on how to install and set up MUI (Material-UI) in a React project. (A)

Lab 10:

- Create a page showing all typography variants (h1–h6, body1, body2, subtitle1, subtitle2, caption). (A)
- Display a heading and paragraph using custom colors and font weights. (A)
- Create a promotional banner text. (A)
- Demonstrate responsive typography. (A)
- Create a Card with custom sx: (A)
 - padding
 - border
 - borderRadius
 - hover shadow
- Create a Button With SX Hover Animation. (B).
- Create Custom Theme (C)

Lab 11: Practicing Table, Tooltip, ImageList in MUI.

- Create a simple table with 3 columns: name, email, role. (A)
- Create a Striped Table Rows (Alternate Colors). (A)
- Create a student table with sortable columns: (B)
 - Sort by Name
 - Sort by Marks
- Create a student table With Pagination (C)
- Set Basic Tooltip on Button. (A)
- Set Tooltip with Transition. (B)

- Implement Pinterest-like layout to display images. (C)

Lab 12: Implementing Layout components in MUI.

- Create a full-page layout: (A)
 - Header
 - Content Area
 - Footer
- Design Two-Column Layout Using Grid. (A)
- Implement Responsive Grid layout. (A)
- Implement Horizontal and Vertical Stack. (B)
- Create 4 boxes: (B)
 - Stats
 - Chart
 - Recent Orders
 - Notifications

Each inside a Paper with padding and elevation.

- Create a top navigation bar with: (B)
 - Logo
 - Search bar
 - Icon buttons
- Implement Permanent Sidebar Layout Using Drawer. (C)

Lab 13: Implementing navigation components in MUI.

- Create a basic AppBar with: (A)
 - App name title
 - Toolbar
 - Center aligned text
- Create an AppBar layout: (B)
 - Logo (Typography)
 - Navigation buttons: Home, About, Contact
 - Using Button inside Toolbar
- Inside the AppBar include: (C)
 - Search input (InputBase)
 - Notifications icon
 - Account icon
 - Flexible spacing using Box

- Create a hamburger menu icon in AppBar that opens a temporary drawer. (C)
- Implement Collapsed drawer that expands on hover, like Gmail sidebar. (C)
- Button opens dropdown: (B)
 - Profile
 - Settings
 - Logout
- Create 3 tabs: (B)
 - Dashboard
 - Users
 - Settings

Content changes on tab selection.

- Display Breadcrumbs Navigation. (A)
- Create bottom navigation with 4 items: (B)
 - Home
 - Search
 - Favorites
 - Profile

Lab 14: Creating a form and validate with formik and yup.

- Implement Basic Login Form (email + password). (A)
- Implement Registration Form with Confirm Password. (A)
- Create a form with: (B)
 - firstName
 - lastName
 - phone

Rules:

 - phone: exactly 10 digits
 - firstName: only letters → `.matches(/^[A-Za-z]+$/)`
 - Use `helperText` inside Material UI `<TextField>`
- Create a student registration form with following field and validate using formik and yup. (B)
 - StudentFirstName (required)
 - StudentLastName
 - StudentEmail (should be email)
 - StudentMobile (should be 10 digits, required)
 - StudentSemester (must be between 1 to 8)
 - StudentBranch (should be from CSE, MECH, CIVIL)

- StudentEnrollment (must be 12 digits, required)
- StudentHobbies (checkbox at least one should be select)

Lab 15: Practicing feedback components in MUI.

- Create a button that shows a Snackbar: (A)
 - Message: "Form submitted successfully!"
 - Auto-hide after 3 seconds
 - Use Snackbar with open, onClose, autoHideDuration
- Show a colored alert inside Snackbar (Practice all 4 severities: success, info, warning, error) (A)
- Create 4 buttons that show Snackbar in: (B)
 - top-left
 - top-right
 - bottom-left
 - bottom-right
- Create a dialog for delete confirmation: (A)
 - "Are you sure you want to delete?"
 - Buttons: Cancel, Delete
- Create a dialog containing a form with: (B)
 - Name
 - Email
 - Submit button
- Create a fullscreen dialog: (B)
 - AppBar with close icon
 - Content area
 - Perfect for mobile screens
- Create loading screen: (A)
 - CircularProgress centered on screen
 - LinearProgress at top
 - Toggle visibility using a button
- Create a card: (C)
 - Image
 - Title
 - Description

Replace content with Skeleton when loading.
- Create a Rating component: (B)
 - Display selected rating value
 - Allow half-star rating

Lab 16: Creating static crud operation using MUI Data Grid.

- Create an app to perform crud operation on a static array containing students data using MUI Data Grid. (A)

Lab 17: Exploring built-in hook in React: useState, useEffect, useContext

- Create a counter with “+” and “-” buttons. The counter value must update using useState. (A)
- Show/Hide a paragraph using a toggle button that changes state true/false. (B)
- A text box updates a <p> showing the typed value. (C)
- Call a public API (e.g., JSONPlaceholder or mockapi) and display list of users once on component load. (A)
- Increase a number (state) every second using setInterval and display on screen. (B)
- Provide a theme string and consume it in multiple child components. (A)
- Store logged-in user details (name + email) in context, display them in child components. (B)
- Save selected language in context, and show messages based on the selected language. (C)

Lab 18: Creating a custom hook in React, also use useDebugValue hook

- Create a useToggle (custom) hook which return a Boolean flag and a function to toggle the state of the flag. (A)
- Create a custom hook to fetch api, return data, error and is_loading state form the hook. (B)
- Use useDebug hook in above hook to debug the values inside custom hook. (B)

Lab 19: Exploring built-in hook in React: useCallback, useMemo, useRef, useReducer

- Write a code in which parent component passes a memoized callback to a child that prints a message on click. (A)
- Write a code which Add items using memoized addItem handler passed to child component from the parent component, make sure child does not re-render unnecessarily. (A)
- Calculate factorial of a number (stored as a state) using useMemo. (A)

- Write a code which Compute `fullName = firstName + lastName` only when either of them is changed. (B)
- Write a program which reads a value of an input text without controlled component (i.e. using `useRef`). (A)
- Write a program which focuses a specific input text which clicked on a button. (A)
- Write a program which Add, toggle, and remove todo items using reducer. (A)

Lab 20: Exploring built-in hook in React: `useDeferredValue`, `useTransition`, `useFormStatus`, `useImperativeHandle`

- Write a code which filters large array based on search; use deferred value to show smooth typing. (A)
- Write a code in which main text updates immediately; preview text updates slowly using `useDeferredValue`. (B)
- Write a code to demonstrate filtering huge list with `startTransition` to prevent UI blocking. (A)
- Write a program which while Switching tabs shows “Loading...” using `useTransition`. (B)
- Write a program which disable submit button and show “Submitting...” while form sends data. (A)
- Write a program which display success message from server after form submission. (B)
- Write a program in which A child input exposes a `.focus()` method to parent using `useImperativeHandle`. (A)
- Write a program in which child component exposes play/pause functions for a video player. (C)

Lab 21: Implement local and global state management in React.

- Build a login form using `useState` to store email and password. Display entered data on submit. (A)
- Create an input field and an “Add Task” button to show tasks in a list. Add delete functionality. (B)
- Create a `UserContext` that stores name & email. Display data in multiple components. (A)
- Implement `ThemeContext` to toggle theme for entire application layout.
- Implement login/logout functionality using Context API. (B)
- Implement Cart functionality with Context API. (C)

Lab 22: Implement centralized state management in React using Redux.

- Use redux to switch theme in an application (A)
- Implement User authentication functionality using redux. (B)
- Implement Cart functionality using redux. (C)

Lab 23: Complete CRUD operation on a data stored in an array using redux

- Write a program which perform CRUD operation on an array stored in store, also implement user authentication in an application. (B)

Lab 24: Implement RTK Query to do caching of API calls.

- Perform CRUD operation on an MockAPI using RTK Query. (A)
- Demonstrate the caching mechanism of RTK Query. (A)

Lab 25: Implement basic chat application using socket.io

Lab 26: Implement complete chat application with backend support to create clone of whatsapp.

Lab 27: Practicing Jest to perform automated testing

- Write a function add(a, b) and test using Jest: (A)
 - correct sum
 - negative numbers
 - zero inputs
- Write a function add(a, b) and test using Jest: (A)
 - correct sum
 - negative numbers
 - zero inputs
- Create a simple <Counter /> component an perform following tests using Jest. (B)
 - renders text
 - button click increments value
- Create useToggle(initialValue) function an perform following tests using Jest. (B)

- initial value
- toggle changes state

Lab 28: Implement Jest on project to perform automated testing.

Lab 29: Practicing component testing and E2E testing

- Perform component testing on project. (A)
- Perform E2E testing on project using Cypress. (A)
- Perform E2E testing on project using Playwright. (A)

Lab 30: Revision Lab