

React & Tailwind Assignment: Interactive Dashboard

Objective

Create a responsive dashboard using React and Tailwind CSS that displays various data visualizations and includes interactive components. The project should demonstrate proficiency in React, Tailwind CSS, state management, and version control.

Requirements

1. Dashboard Layout

- Create a responsive layout with a sidebar and main content area
- Implement a collapsible sidebar for mobile views
- Use Tailwind CSS for styling

2. Data Visualizations

- Implement at least three different types of charts (e.g., line chart, bar chart, pie chart) using a charting library of your choice (e.g., Recharts, Chart.js)
- Make the charts responsive and interactive (e.g., show tooltips on hover)

3. Interactive Components

- Create a data table with sorting and filtering capabilities
- Implement a search functionality that filters data across all components
- Add a date range picker that updates the displayed data

4. State Management

- Use React hooks for local state management
- Implement context API or Redux for global state management

5. API Integration

- Fetch data from a mock API (you can use tools like json-server or Mirage JS to create a mock backend)
- Implement error handling and loading states

6. Performance Optimization

- Implement code splitting and lazy loading for dashboard components
- Optimize render performance using `React.memo` or `useMemo` where appropriate

7. Testing

- Write unit tests for key components using Jest and React Testing Library
- Achieve at least 70% test coverage

8. Version Control

- Use Git for version control
- Create a well-structured Git history with meaningful commit messages
- Implement feature branching and pull requests

9. Documentation

- Provide a `README.md` file with clear instructions on how to set up and run the project
- Include inline code comments explaining complex logic or implementations

Bonus Points

- Implement a dark mode toggle using Tailwind CSS
- Add animations and transitions to enhance user experience
- Create a CI/CD pipeline using GitHub Actions or similar tools

Evaluation Criteria

- Code quality and organization
- Effective use of React hooks and Tailwind CSS
- Responsiveness and cross-browser compatibility
- User experience and interface design
- Test coverage and quality
- Git commit history and branch management
- Documentation clarity

Submission

- Provide a link to the GitHub repository containing your project
- Include a live demo link (e.g., hosted on Netlify, Vercel, or GitHub Pages)
- Submit a brief write-up explaining your approach, challenges faced, and how you overcame them

Good luck!