

Assignment : Responsive Dashboard with News and Payout Features

Responsibilities:

1. User Authentication:

- Implement secure **login functionality** using email-password authentication or third-party tools (e.g., Google, GitHub OAuth).

2. News and Blog Data Integration:

- Fetch data from **free third-party news APIs** (e.g., News API, Guardian API).
- Display article/blog counts with details like author, date, and type.

3. Filtering and Search:

- Implement filters to allow users to search by:
 - **Author**
 - **Date range**
 - **Type** (e.g., news, blogs)
- Add a **global search bar** for quick keyword searches.

4. Responsive Design:

- Build a fully **responsive UI** compatible with both mobile and desktop devices.

5. Payout Calculator:

- Allow admins to set a **payout per article/blog** value.
- Store payout data in **local storage**.
- Automatically calculate total payouts based on the number of articles/blogs and their rates.

6. Export Functionality:

- Enable users to export payout reports as:
 - **PDF**
 - **CSV**
 - **Google Sheets** integration.

7. Frontend Frameworks & Tools:

- Use modern frameworks like NextJs for development.
 - Ensure proper state management using tools like **Redux**, **Pinia**, or **Context API**.
-

Key Features to Implement:

1. Dashboard:

- Overview of total articles/blogs fetched from the API.
- Visual representation of payouts and filters.

2. News Analytics:

- Graphical charts (e.g., bar, pie, or line charts) showing article trends by author or type.

3. Payout Details:

- A simple table listing authors, articles, and calculated payouts.
- Inline editing for payout rates.

4. Export Options:

- Export filtered or complete data in **PDF/CSV/Google Sheets** format.

5. Error Handling:

- Graceful fallback for API failures (e.g., display a message if the news API is unreachable).
-

Technical Skills Required:

- **Frontend Frameworks:** React.js, Next.js,
 - **APIs and Integration:** Fetching and handling data from third-party REST APIs.
 - **Responsive Design:** Expertise in building mobile-first, responsive UIs using CSS or frameworks like Tailwind CSS.
 - **State Management:** Redux, Recoil or Context API.
-

Bonus Points:

- Implementing offline mode with **IndexedDB** or **localStorage**.
 - Incorporating dark mode for the dashboard.
 - Ensuring high Lighthouse scores for performance, accessibility, and SEO.
-