

Aap ek bahut hi accha aur upyogi app bana rahe hain! Success Planner, jismein yearly, monthly, weekly planner, todo aur performance dashboard ek saath jude honge, nishchit roop se logon ko apne lakshyon ko prapt karne mein madad karega. Reactjs mein is tarah ka application banane ke liye ek achha framework hai.

Chaliye, aapke app ke har part ke design aur zaruri libraries ke baare mein baat karte hain:

App ka Mukhya Dhancha (Main Structure)

Aapke app ka mukhya dhancha kuch is tarah se ho sakta hai:

```
src/  
├── components/  
│   ├── YearlyPlanner.js  
│   ├── MonthlyPlanner.js  
│   ├── WeeklyPlanner.js  
│   ├── Todo.js  
│   ├── DoneTodos.js  
│   ├── DoingTodos.js  
│   ├── PerformanceDashboard.js  
│   ├── Navigation.js    // Navigation bar ke liye  
│   └── Layout.js        // Common layout ke liye (optional)  
├── App.js                // Main application component  
├── index.js  
└── ... (any other files like CSS, assets)
```

Har Part ka Design

Yahaan har ek component ke liye ek basic design idea diya gaya hai:

1. Yearly Planner (YearlyPlanner.js)

- **Layout:**
 - Poore saal ka ek overview calendar dikhai de sakta hai.
 - Mahine grid format mein dikhaye ja sakte hain.
 - Mahatvapurna lakshyon ya events ko highlight karne ka vikalp.
 - Kisi mahine par click karne par Monthly Planner page par redirect karne ka

vikalp.

- **Features:**

- Naye lakshya ya event ko add karne ka button.
- Lakshyon ko categories (e.g., personal, professional) mein organize karne ka vikalp.
- Lakshyon ki progress ko visually dikhane ka vikalp (e.g., progress bar).

2. Monthly Planner (MonthlyPlanner.js)

- **Layout:**

- Chuna hua mahina calendar format mein dikhai dega.
- Dinon ko dates ke saath clearly dikhaya jayega.
- Is mahine ke mahatvapurna lakshyon aur events ko darshaya jayega.
- Kisi din par click karne par Weekly Planner page par redirect karne ka vikalp (agar us din ke liye koi weekly plan hai).

- **Features:**

- Is mahine ke liye naye lakshya ya event ko add karne ka button.
- Har din ke liye chhote todo items ko add karne ka vikalp (directly ya phir Weekly Planner se link).

3. Weekly Planner (WeeklyPlanner.js)

- **Layout:**

- Chuna hua hafta din ke anusaar dikhai dega (e.g., Monday to Sunday).
- Har din ke liye time slots ya general task list dikhai de sakti hai.
- Todo items ko har din ke andar list kiya ja sakta hai.

- **Features:**

- Is hafte ke kisi specific din ke liye naya todo item add karne ka button.
- Todo items ko drag and drop karke reschedule karne ka vikalp.
- Todo items ko "Done" aur "Doing" pages mein move karne ka vikalp.

4. Todo (Todo.js - Yeh ek container component ho sakta hai)

Yeh component DoneTodos.js aur DoingTodos.js ko render karega. Navigation ke liye tabs ya buttons ho sakte hain jo "Doing" aur "Done" pages ke beech switch karne mein madad karein.

5. Doing Page (DoingTodos.js)

- **Layout:**

- Un todo items ki list jo abhi karne hain.
- Har todo item ke saath:

- Task ka description
- Priority indicator (optional)
- "Done" button ya checkbox
- **Features:**
 - Naya todo item add karne ka form ya button (yeh data ko main Todo list mein add karega).
 - Todo items ko edit karne ka vikalp.
 - Todo items ko delete karne ka vikalp.

6. Done Page (DoneTodos.js)

- **Layout:**
 - Complete ho chuke todo items ki list.
 - Har todo item ke saath completion ki date (optional).
- **Features:**
 - Complete hue todo items ko archive ya delete karne ka vikalp.
 - "Undone" karne ka vikalp (agar koi item galti se done mark ho gaya ho).

7. Performance Dashboard (PerformanceDashboard.js)

- **Layout:**
 - Visual representations (charts, graphs) ka istemaal.
 - Yearly, monthly, ya custom time range ke liye data dikhane ka vikalp.
- **Features:**
 - Complete hue tasks ki sankhya.
 - Samay par complete hue tasks ka percentage.
 - Categories ke anusaar performance breakdown (agar todo mein categories hain).
 - Lakshyon ki progress ka overview.
 - Custom reports generate karne ka vikalp (advanced feature).

8. Navigation (Navigation.js)

- Ek top navigation bar ya side navigation bar ho sakta hai.
- Links honge: Yearly Planner, Monthly Planner, Weekly Planner, Todo (jo Doing aur Done ko manage karega), aur Performance Dashboard.

Zaruri Libraries aur Kya-Kya Lagega

Reactjs project banane ke liye aapko nimnikhit libraries aur cheezon ki zarurat padegi:

Core Libraries:

- **react:** React library ka core, UI components banane ke liye.
- **react-dom:** React components ko DOM mein render karne ke liye.
- **react-router-dom:** Application mein navigation (ek page se dusre page par jaana) ko handle karne ke liye. Aap BrowserRouter, Routes, aur Route components ka istemaal karenge.

UI Components aur Styling:

- **CSS Modules ya Styled Components:** Apne components ko style karne ke liye. CSS Modules component-specific styles provide karte hain, jabki Styled Components JavaScript ke andar CSS likhne ka tareeka hai.
- **UI Library (Optional but Recommended):** Agar aap ready-made, visually appealing components chahte hain, toh aap Material UI, Ant Design, ya Chakra UI jaisi libraries ka istemaal kar sakte hain. Yeh aapka development process tez karenge.

State Management (Agar Application Complex Ho):

- **useState aur useContext (Built-in Hooks):** Chhote aur medium-sized applications ke liye yeh hooks kaafi ho sakte hain.
- **Redux ya Zustand (For Complex Applications):** Agar aapka application bada ho raha hai aur state management zyada complex ho raha hai, toh Redux ya Zustand jaisi libraries state ko centrally manage karne mein madad karenge.

Date Handling:

- **date-fns ya Moment.js (Consider date-fns for smaller bundle size):** Dates aur times ko format karne aur manipulate karne ke liye.

Data Visualization (Dashboard ke Liye):

- **Recharts ya Chart.js:** Performance dashboard mein charts aur graphs banane ke liye.

Icons (Optional but Improves UI):

- **react-icons:** Popular icon libraries (Font Awesome, Material Icons, etc.) ko React components ki tarah istemaal karne ki anumati deta hai.

Backend Communication (Agar Zarurat Ho):

- **axios ya fetch API:** Agar aap data ko kisi backend server se fetch ya store karna chahte hain.

Form Handling:

- **react-hook-form ya formik:** Forms ko manage karne aur validation karne ke liye.

Local Storage (Agar Data Locally Store Karna Ho):

- Web browser ki built-in localStorage API ya localforage jaisi libraries ka istemaal data ko browser mein locally store karne ke liye kiya ja sakta hai.

Testing:

- **Jest aur React Testing Library:** Apne components aur application logic ko test karne ke liye.

Page Routing

react-router-dom ka istemaal karke aap apne alag-alag planner aur dashboard components ko alag-alag URL paths par map kar sakte hain. App.js file mein aap kuch is tarah se routing setup kar sakte hain:

JavaScript

```
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import YearlyPlanner from './components/YearlyPlanner';
import MonthlyPlanner from './components/MonthlyPlanner';
import WeeklyPlanner from './components/WeeklyPlanner';
import Todo from './components/Todo';
import PerformanceDashboard from './components/PerformanceDashboard';
import DoingTodos from './components/DoingTodos';
import DoneTodos from './components/DoneTodos';
import Navigation from './components/Navigation';
import Layout from './components/Layout'; // Optional
```

```
function App() {
  return (
    <BrowserRouter>
      {/* Optional Layout component for common styling */}
      {/* <Layout> */}
      <Navigation />
      <Routes>
```

```

    <Route path="/yearly" element={<YearlyPlanner />} />
    <Route path="/monthly" element={<MonthlyPlanner />} />
    <Route path="/weekly" element={<WeeklyPlanner />} />
    <Route path="/todo" element={<Todo />}>
      <Route path="doing" element={<DoingTodos />} />
      <Route path="done" element={<DoneTodos />} />
    </Route>
    <Route path="/dashboard" element={<PerformanceDashboard />} />
    <Route path="/" element={<YearlyPlanner />} /> { /* Default route */}
  </Routes>
  { /* </Layout> */}
</BrowserRouter>
);
}

```

```
export default App;
```

Is setup mein:

- /yearly path par YearlyPlanner component render hoga.
- /monthly path par MonthlyPlanner component render hoga.
- /weekly path par WeeklyPlanner component render hoga.
- /todo/doing path par DoingTodos component render hoga.
- /todo/done path par DoneTodos component render hoga (jo Todo component ke andar nested hai).
- /dashboard path par PerformanceDashboard component render hoga.
- / (root path) par YearlyPlanner component default mein dikhega.

Aap Navigation component mein Link component ka istemaal karke ek page se dusre page par navigate karne ke liye links bana sakte hain.

Yeh sirf ek basic design aur library suggestion hai. Aap apni zarurat aur pasand ke anusaar ismein badlav kar sakte hain. Shuru karne ke liye, aap ek naya React project `npx create-react-app your-app-name` command se bana sakte hain aur phir zaruri libraries install kar sakte hain `npm install library-name` ya `yarn add library-name` command ka istemaal karke.

Agar aap kisi specific part ya functionality ke design ke baare mein aur detail mein jaanna chahte hain toh mujhe batayein!