Therapist Finder Prototype - 30 Hour Hackathon Challenge

Project Overview

Build a single-page application (SPA) that allows users to find and filter mental health professionals in Pakistan. The application should provide an intuitive interface for discovering therapists based on various criteria from the provided dataset.

Technical Requirements

Recommended Stack (You may use alternatives)

- **Frontend**: React, Next.js, Vue, Angular, or Svelte

- **Backend**: Node.js, Python (Django/FastAPI), Ruby on Rails, or any backend framework

- **Database**: PostgreSQL, MySQL, MongoDB, or SQLite

- **Styling**: Tailwind, CSS Modules, Styled Components, or any CSS solution

- **Language**: TypeScript recommended but JavaScript is acceptable

- **Deployment**: Any platform (Vercel, Netlify, Railway, Render, etc.)

Why We're Flexible on Stack

- We want to see you work with tools you're comfortable with

- Your problem-solving approach matters more than specific framework knowledge

- Code quality and architecture decisions are what we're evaluating

- Show us your best work with your preferred tools

Core Features to Implement

1. Data Management (3 hours)

- Import the CSV data into your chosen database

- Design an appropriate schema for the therapist data

- Ensure data integrity and handle missing values

2. Backend API (6 hours)

Create RESTful or GraphQL endpoints for:

- List all therapists with pagination

- Search and filter therapists

- Get individual therapist details
- Get filter options with counts (for UI)

3. Frontend Application (15 hours)

UI/UX Design Requirements

Main Application Layout

Header Section

- **Logo/Brand**: " 🗨 MindCare Pakistan" on the left
- **Search Bar**: Central search input with search button
- **Responsive**: Collapses to mobile menu on smaller screens

Two-Column Layout Structure

Left Column - Filter Panel (20% width on desktop, full width drawer on mobile)

City Filter Section

- Checkbox list with cities
- Options: Karachi, Lahore, Islamabad, Other
- Show count next to each option

Experience Filter Section

- Radio button group
- Options: 0-5 years, 5-10 years, 10-15 years, 15+ years

Gender Filter Section

- Checkbox options: Male, Female
- Display count for each

Fee Range Filter Section

- Radio button group
- Options: Under Rs.2000, Rs.2000-4000, Rs.4000-6000, Above Rs.6000

Consultation Mode Filter

- Checkboxes: In-person, Online

- Clear All button at bottom

Right Column - Results Grid (80% width on desktop, full width on mobile)

Therapist Card Component Each card should display:

- Placeholder for photo/avatar
- Doctor name (prominent)
- Rating (if available)
- City location with icon
- Fee amount with currency
- "View Details" button

Grid Layout

- 3 columns on desktop
- 2 columns on tablet
- 1 column on mobile
- Pagination or "Load More" at bottom
- Results count display

Therapist Detail View (Modal or New Page)

Header Section

- Close button (if modal)
- Doctor photo/avatar placeholder
- Name as main heading
- Gender and experience years as subtitle

Contact Information Bar

- Location with map icon
- Fee with money icon
- Phone with phone icon (clickable)
- Email with email icon (clickable)

Content Sections (Tabbed or Accordion)

1. **Education Section**

- Display education history from CSV

- Format as bulleted list or timeline

2. **Experience Section**
   - Work history and positions

   - Years of practice

3. **Expertise Section**
   - List of specializations

   - Treatment approaches

4. **About Section**
   - Personal statement/bio

   - Philosophy or approach

5. **Consultation Modes**
   - Available modes (in-person/online)

   - Timing if available

Action Buttons (Footer)

- Primary: "Call Now" button

- Secondary: "Send Email" button

- Tertiary: "Visit Profile" link

Mobile Responsive Considerations

Mobile-First Design Requirements

- Filter panel as slide-out drawer with hamburger menu

- Search bar below header

- Single column card layout

- Sticky filter button at bottom

- Touch-friendly button sizes (min 44x44px)

- Swipeable cards on mobile

- Bottom sheet for therapist details

Breakpoints

- Mobile: < 768px

- Tablet: 768px - 1024px

- Desktop: > 1024px

Visual Design Guidelines

Color Scheme Suggestions

- Primary: Teal/Turquoise (🔵 #0891b2) - Medical/wellness association

- Secondary: Warm Gray (⬤ #64748b) - Professional

- Accent: Coral (🔴 #fb7185) - Call-to-action

- Background: Light gray (⚪ #f8fafc) or white

- Success: Green (🟢 #10b981)

- Error: Red (🔴 #ef4444)

Typography

- Headers: Bold, sans-serif (e.g., Inter, Poppins)

- Body: Regular weight, good readability

- Card titles: Semi-bold, larger size

- Minimum font size: 14px on mobile

Component Styling

- Cards: White background with subtle shadow

- Rounded corners (8px radius)

- Hover effects on interactive elements

- Loading skeletons for async content

- Smooth transitions (300ms)

Filter Specifications

Based on the CSV data, implement these filters:

1. **City Filter** (Multi-select)
   - Extract unique cities from the `city` column

   - Show count of therapists per city

   - Include "Other" option for less common cities

2. **Experience Range** (Single-select)

- 0-5 years (Entry Level)

- 5-10 years (Mid Level)

- 10-15 years (Senior)

- 15+ years (Expert)

3. **Gender Filter** (Multi-select)

- Male

- Female

- Show count for each

4. **Fee Range** (Single-select)

- Under Rs. 2,000

- Rs. 2,000 - 4,000

- Rs. 4,000 - 6,000

- Above Rs. 6,000

- Parse from `fee_amount` column

5. **Consultation Mode** (Multi-select)

- In-person

- Online

- Both

- Parse from `modes` column

6. **Search Bar** (Text input)

- Search across: name, expertise, education, about

- Implement debouncing (300ms)

- Highlight matching terms in results (optional)

Required Features (Must Have)

1. **Core Functionality**

- All filters working correctly

- Search functionality

- Individual therapist detail view

- Responsive design (mobile & desktop)

2. **Performance**

- Page load time < 3 seconds

- Smooth filtering without page refresh

- Handle 79 records efficiently

3. **User Experience**
   - Loading states for async operations

   - Error handling for failed requests

   - Empty states when no results

   - Clear filter indicators

4. **Data Handling**
   - Handle missing data gracefully

   - Clean data presentation

   - Accurate filter counts

Bonus Features (Nice to Have)

- URL state persistence (shareable filter links)

- Export filtered results

- Sort options (by fee, experience, name)

- Animations and transitions

- Dark mode

- Advanced search (fuzzy matching)

- Analytics dashboard

- Favorites/bookmarking

- Recently viewed therapists

Evaluation Criteria

Problem Solving & Logic (30%)

- How you handle edge cases

- Data processing approach

- Algorithm efficiency

- Architecture decisions

Code Quality (25%)

- Clean, readable code

- Proper separation of concerns

- Reusable components/functions

- Error handling

- Comments where necessary

UI/UX Design (25%)

- Visual appeal

- Intuitive navigation

- Responsive design

- Accessibility considerations

- Overall user experience

Completeness & Functionality (20%)

- All required features working

- Data accuracy

- Performance metrics met

- Bonus features attempted

Submission Requirements

1. **GitHub Repository** containing:
   - README.md with:
     - Setup instructions

     - Technology choices explanation

     - Architecture decisions

     - Known issues/limitations

   - Environment variables template

   - Database schema/structure

2. **Live Demo URL**
   - Deployed application

   - Sample data loaded

3. **Short Video or Screenshots** (2-3 minutes)
   - Demonstrating all filters

   - Showing responsive design

- Highlighting special features

What We're Looking For

Technical Skills

- Can you build a working application from scratch?

- Do you understand database design and querying?

- Can you create an intuitive user interface?

- How do you handle state management?

Problem-Solving

- How do you parse and clean messy data?

- How do you optimize for performance?

- How do you handle edge cases?

Code Quality

- Is your code maintainable?

- Would another developer understand your code?

- Did you follow best practices for your chosen stack?

Product Thinking

- Did you consider the end user?

- Are there thoughtful UX decisions?

- Did you prioritize the right features?

Tips for Success

1. **Choose Your Comfort Zone**: Use technologies you know well

2. **Start Simple**: Get a working version first, then enhance

3. **Focus on User Experience**: Make it intuitive and pleasant to use

4. **Handle Edge Cases**: What if data is missing? What if no results?

5. **Document Your Decisions**: Explain why you chose certain approaches

6. **Test Your Application**: Ensure all features work before submission

Time Management Suggestion

- **Hour 1-3**: Setup, database design, data import

- **Hour 4-9**: Backend development

- **Hour 10-20**: Frontend development

- **Hour 21-25**: Testing and bug fixes

- **Hour 26-28**: Polish and bonus features

- **Hour 29-30**: Documentation and deployment

Data Processing Notes

From the provided CSV:

- 79 therapist profiles

- Fields include: name, profile_url, gender, city, experience_years, email, phone, modes, education, experience, expertise, about, fees

- Some fields may have missing values

- `fees_raw` needs parsing to extract numeric values

- `modes` field needs parsing for consultation types

- Some therapists have multiple emails in `emails_all`

Questions?

If you have questions about the requirements, make reasonable assumptions and document them in your README.

Good luck! We're excited to see what you build!