KIXAR - React Native + Full-Stack Internship Assignment

DEADLINE: 26 NOV 11:59PM (strictly)
Estimated Effort: 4-6 focused hours

Objective

Build a **React Native mini-application** that:

1. Replicates the UI screens provided in this Figma link:
   ⊡ Figma
2. Implements a **functional booking flow** using static/demo data.
3. Demonstrates clarity, structure, reusability, and strong command of React Native fundamentals.

Deliverables (3 Screens)

1. Turf Details Screen (from Figma)

UI Requirements

Replicate all visible components from the Figma design, including:

- Banner Image
- Turf name, rating, address
- "Get Directions" + Call button
- About section
- Timing information (static/collapsible)
- Amenities list
- Sports availability
- Offers section
- Reviews section (static demo data)
- Map preview
- **Sticky bottom bar** with:
  - Price per hour

- ○ "Book Now" CTA

Functionality

- "Book Now" → Navigates to Booking Page
- Use static JSON for turf details

2. Booking Page (from Figma)

UI Requirements

Match the layout as per the Figma reference:

- Date selector
- Time period options (Morning / Noon / Evening / Twilight)
- Timeline slot area
- Court selection (Court A / Court B)
- Player count selector (+/-)
- Pricing section
- "Next" button

Functionality

- Date + slot selection is mandatory
- "Next" must remain disabled until valid
- On "Next" → Save booking into global app state (Context API)

3. My Bookings Screen

UI Requirements

- List of saved bookings:
  - ○ Turf name
  - ○ Chosen date
  - ○ Chosen slot
  - ○ Court selected
  - ○ Player count

○ Price/hour
● Empty state if no bookings

Functionality

● Bookings load from global state
● Optional: persist in AsyncStorage

Technical Requirements

● React Native (Expo preferred)
● Hooks only
● React Navigation
● Context API for booking state
● Static data (no backend required)
● Clean folder structure:
    ○ /components
    ○ /screens
    ○ /context
    ○ /data
    ○ /assets

Evaluation Levels

Level 1 - Minimum Qualification

● Fully working navigation
● Correct param passing
● Working booking state
● Correct button validation
● My Bookings reflects stored bookings

Level 2 - Recommended

● AsyncStorage
● Reusable UI components
● Derived UI values

Level 3 - Bonus

- Animations
- TypeScript
- Filtering or category logic

Submission Requirements

GitHub Repository

Your repository must include:

- Full source code
- README with:
    - Setup Instructions
    - Assumptions
    - Time spent
    - Known issues
    - Pending improvements

Optional Extra

- 1-2 minute walkthrough video
- Live Expo Snack link

How to Submit
Submit your assignment at: https://forms.gle/J5smKU5aHt5WQoGv8

Evaluation Criteria

Category	Weight
UI Accuracy	30%
Functionality	40%
Code Quality	20%
Enhancements	10%

Good luck - build fast, clean, and user-centric!
Welcome to Kixar's engineering culture.