Detailed Requests for Refactoring Etoile Yachts Mobile Application to Align with Information Architecture and Mental Model

Request

Please implement the refactoring process for the Etoile Yachts mobile application by aligning the app's structure, functionalities, and database models with the updated information architecture and mental model. The focus is on achieving seamless user navigation, modular backend services, and robust data management.

Frontend Requests

1. Refactor Navigation Structure

- Implement a **Bottom Tab Navigator** for primary sections: Home, Explore, Bookings, Profile, Support, Notifications.
- Use a **Stack Navigator** for each section's detailed pages.
- Add dynamic highlighting for active tabs.

Dependencies:

- `react-navigation/native`
- `react-native-screens`
- `react-native-safe-area-context`

2. Component Hierarchy Refactoring

- Build reusable components for:
  - **Cards** (e.g., activity, yacht details, booking summaries).
  - **Filters** (e.g., location, price, preferences).
  - **Forms** (e.g., registration, booking details).
  - **Buttons and Inputs** (consistent design system).

Dependencies:

- `react-native-elements` or `react-native-paper` for consistent UI components.

3. Implement State Management

- Centralize state management using:

    - `redux` for global state (e.g., user data, bookings).
    - `redux-thunk` for asynchronous API calls.

Dependencies:

- `redux`
- `react-redux`
- `redux-thunk`

4. Integrate Search and Filters

- Add a **search bar** with autocomplete functionality for activities, yachts, and locations.
- Implement dynamic filters:
    - Price range slider.
    - Location picker using Google Maps API.
    - Activity types and availability toggles.

Dependencies:

- `react-native-maps`
- Google Places API for location autocomplete.

5. Update Profile Management Pages

- Separate views for Peer Consumers, Peer Producers, and Partners.
- Include editable sections for:
    - Personal information.
    - Preferences (for consumers).
    - Service offerings (for producers and partners).

6. Notifications System

- Create a centralized notification feed.
- Categorize notifications (e.g., booking updates, promotions, reminders).

Dependencies:

- Firebase Cloud Messaging (FCM).

Backend Requests

1. Update Database Schema

- Redesign tables to match the updated IA:

Tables for Peer Consumers:

- `users`

    - `id` (Primary Key)
    - `name`
    - `email`
    - `phone`
    - `profile_picture`
    - `loyalty_points`
- `preferences`
    - `id` (Primary Key)
    - `user_id` (Foreign Key to `users`)
    - `activity_preferences`
    - `dietary_restrictions`
    - `accessibility_needs`

Tables for Peer Producers and Partners:

- `producers`
    - `id` (Primary Key)
    - `name`
    - `business_name`
    - `email`
    - `phone`
    - `certifications`
- `services`
    - `id` (Primary Key)
    - `producer_id` (Foreign Key to `producers`)
    - `type` (e.g., yacht rental, catering)
    - `details`
    - `availability_schedule`

Shared Tables:

- `bookings`
    - `id` (Primary Key)
    - `user_id` (Foreign Key to `users`)
    - `service_id` (Foreign Key to `services`)
    - `status` (e.g., pending, confirmed)
    - `price`
- `notifications`
    - `id` (Primary Key)
    - `user_id` (Foreign Key to `users`)
    - `content`
    - `type` (e.g., booking, promotion)
    - `timestamp`

Dependencies:

- PostgreSQL for relational data.
- Sequelize or Prisma for ORM.

2. Update API Endpoints

- Modularize endpoints based on user roles:

Consumer Endpoints:

- `GET /explore`: Fetch yachts, activities, and packages.
- `POST /bookings`: Create a booking.
- `GET /bookings/:id`: Retrieve booking details.

Producer Endpoints:

- `POST /services`: Add or update service details.
- `GET /services`: Retrieve services offered.

Shared Endpoints:

- `POST /login`: Authenticate users.
- `GET /notifications`: Fetch user notifications.
- `POST /feedback`: Submit user feedback.

Dependencies:

- `express` for API development.
- `jsonwebtoken` for authentication.

AI Integration Requests

1. Recommendation System

- Implement AI to recommend:
  - Activities based on user preferences and booking history.
  - Packages based on current trends and availability.

Dependencies:

- Python-based recommendation engine integrated via REST API.

2. Dynamic Pricing System

- Adjust prices dynamically based on demand, seasonality, and resource availability.

Dependencies:

- AI pricing model trained on historical booking data.

Testing and Deployment Requests

1. Automated Testing

- Implement unit tests for all components and backend endpoints.
- Use end-to-end tests for critical workflows (e.g., booking, profile updates).

Dependencies:

- `jest` and `react-testing-library` for frontend testing.
- `mocha` and `chai` for backend testing.

2. Deployment Setup

- Use Replit's deployment pipeline to host the app's frontend and backend.
- Integrate CI/CD for automated builds and testing.

Dependencies:

- Docker for containerization.
- GitHub Actions for CI/CD.

Final Deliverables for Refactor

1. Updated React Native frontend aligned with new IA.
2. Backend services with modularized endpoints and refactored database schema.
3. AI-powered features for recommendations and dynamic pricing.
4. Fully tested and deployed application ready for user engagement.

This comprehensive plan ensures alignment with the Etoile Yachts mental model, delivering a robust and user-centric mobile application.