Mobile App Development Task

Overview

Create a mini mobile application using Expo React Native with a Node.js backend. This task evaluates your full-stack development skills including authentication, real-time updates, and mobile app development.

Frontend Requirements (React Native with Expo)

Core Features

- **Authentication System**
    - Login screen with email/password
    - Registration screen for new users
    - "Forgot Password" functionality
    - JWT token-based authentication with expiration handling
    - Automatic logout when token expires
- **Main Application Screen**
    - Display a QR code containing a random UUID
    - QR code updates automatically every 60 seconds
    - Clean, user-friendly interface
    - Proper loading states and error handling

Technical Requirements

- Use Expo SDK (latest stable version)
- Implement proper navigation (React Navigation)
- State management (Context API or Redux)
- Token storage using secure storage
- QR code generation library
- Responsive design for different screen sizes

Backend Requirements (Node.js)

API Endpoints

- `POST /auth/register` - User registration
- `POST /auth/login` - User authentication
- `POST /auth/forgot-password` - Password reset functionality
- `GET /auth/me` - Get current user (protected route)
- `GET /qr/current` - Get current QR code UUID (protected route)

Technical Requirements

- Express.js framework
- JWT authentication with expiration
- Password hashing (bcrypt)
- Input validation and sanitization
- Error handling middleware
- UUID generation for QR codes
- QR code UUID refresh every 60 seconds
- Database integration (MongoDB, PostgreSQL, or SQLite)

Acceptance Criteria

Functionality

- Users can register with email and password
- Users can login and receive a JWT token
- Token expiration is properly handled
- Forgot password sends reset instructions
- QR code displays random UUID after successful login
- QR code automatically updates every 60 seconds
- Proper logout functionality

Code Quality

- Clean, readable code with proper structure
- Error handling for network requests and edge cases
- Proper security practices (no hardcoded secrets, input validation)
- Basic documentation/comments for complex logic

Bonus Points

- Real-time QR code updates using WebSockets
- Email integration for password reset
- Unit tests for critical functions
- Docker configuration
- Environment variable configuration
- Exceptional UI/UX with micro-interactions and delightful animations
- Dark/Light theme support
- Offline handling and network error recovery

Deliverables

1. **Source Code**
    - Frontend: Complete Expo React Native project
    - Backend: Node.js API with all endpoints

    - README.md with setup and run instructions
2. **Documentation**
    - API documentation (endpoints, request/response formats)
    - Setup instructions for both frontend and backend
    - List of dependencies and versions used
3. **Demonstration**
    - Working application (can be demonstrated via screen recording)
    - Test user credentials for evaluation

Evaluation Criteria

- **Functionality** (40%): All features work as specified
- **Code Quality** (30%): Clean, maintainable, and well-structured code
- **User Experience** (20%): Intuitive interface and smooth user flow
- **Security** (10%): Proper authentication and data handling

Time Expectation

This task should take approximately 6-8 hours to complete. Focus on core functionality first, then add bonus features if time permits.

Questions?

If you have any questions about the requirements or need clarification, please don't hesitate to ask.