# Barcode Scanner App Documentation

## Project Overview

The Barcode Scanner App is a mobile application developed using \*\*React Native\*\* (frontend), \*\*Node.js with Express\*\* (backend), and \*\*MongoDB\*\* (database). It offers a streamlined authentication system for users and an admin, with barcode scanning tied to a points-based wallet system. Users can scan and manage barcodes with detailed views, while the admin oversees user approvals, barcode data, and points. The app uses Expo for APK building and emphasizes an attractive UI/UX design with light and dark mode support for a polished mobile experience.

---

## Core Features and Functionality

### 1. Home Page

- \*\*Description\*\*: The initial screen for unauthenticated users.

- \*\*Options\*\*:

- \*\*Login\*\*: Button to navigate to the login screen.

- \*\*Register\*\*: Button to navigate to the registration screen.

- \*\*Selection\*\*: Users choose to log in or register as a \*\*User\*\* or \*\*Admin\*\* via a dropdown or toggle.

- \*\*Persistent Login\*\*: If a user logs in without manually logging out, the app stores credentials locally and automatically displays their dashboard (User or Admin) upon reopening.

### 2. Authentication Screens

- \*\*Login Screen\*\*:

- Fields: Email, Password, Role (User/Admin dropdown).

- Action: Authenticates credentials with the backend and redirects to the appropriate dashboard, persisting the session until manual logout.

- \*\*Register Screen\*\*:

- Fields: Name, Email, Password, Role (User/Admin dropdown), Location (e.g., city or coordinates).

- Action: Submits registration data to the backend.

- For Users: Displays: \*"Your account is pending approval by admin."\* Login is blocked until approved.

- For Admin: Limits to one hardcoded account (e.g., email: `admin@example.com`, password: `admin123`). Rejects additional admin registrations with: \*"Admin account already exists."\*

### 3. User Features

- \*\*User Dashboard\*\*:

- \*\*Barcode Scanning\*\*:

- Button: \*\*"Scan Barcode"\*\*.

- Action: Prompts camera-based barcode scanning.

- Behavior:

- New barcode: Saves to the database with user ID, timestamp, location (via geolocation), and awards points (default: 50, admin-adjustable).

- Existing barcode: Displays: \*"Barcode expired."\*

- \*\*Barcode List\*\*:

- Shows all user-scanned barcodes with:

- Barcode value

- Points earned

- Timestamp

- Location (e.g., "New York" or coordinates)

- \*\*Wallet\*\*:

- Displays total points (e.g., "Points: 150"), updated per new scan based on admin-set value.

### 4. Admin Features

- \*\*Admin Dashboard\*\*:

- \*\*User Management\*\*:

- Lists all users with: Name, Email, Status (Pending/Approved/Disapproved), Points, Location.

- Search by:

- Name (partial/full match).

- Location (e.g., city or proximity).

- Actions per user:

- \*\*Approve\*\*: Activates a pending account.

- \*\*Disapprove\*\*: Deactivates an account (blocks login).

- \*\*Delete\*\*: Removes user and their data.

- \*\*View/Edit Profile\*\*: Shows and allows editing of user details (Name, Email, Location, Points).

- \*\*Reset Points\*\*: Sets points to zero.

- \*\*Barcode Management\*\*:

- Displays all barcodes grouped by user (e.g., "User A: 5 barcodes").

- Details: Value, User, Timestamp, Location, Points Awarded.

- Search by barcode value:

- Found: Shows user who scanned it with profile link.

- Not found: Displays: \*"Barcode not in database."\*

- Actions:

- \*\*Delete Specific Barcode\*\*: Removes one barcode.

- \*\*Delete All Barcodes\*\*: Clears barcodes for a user or all users.

- \*\*Points Management\*\*:

- Set points per scan (default: 50, adjustable).

- View and reset user points.

- \*\*Push Notifications\*\*:

- Sends notifications for:

- Account approval ("Your account has been approved!").

- Account disapproval ("Your account has been disapproved by admin.").

- Points reset ("Your points have been reset by admin.").

- \*\*Barcode Export\*\*:

- Exports all barcodes (global or per user) as CSV or PDF, with download/share options.

### 5. Dark Mode

- \*\*Description\*\*: Toggle between light and dark themes across the entire app.

- \*\*Implementation\*\*: Uses `react-native-paper`’s theming system.

- \*\*Toggle\*\*: A bulb icon (from `react-native-vector-icons`, e.g., `MaterialIcons/lightbulb-outline`) in the header or settings toggles the mode.

- \*\*Visibility\*\*:

- \*\*Light Mode\*\*: White background (`#ffffff`), dark text (`#000000`) for high contrast and readability.

- \*\*Dark Mode\*\*: Dark background (`#121212`), light text (`#ffffff`) for good visibility and reduced eye strain.

- \*\*Impact\*\*: Applies to all screens, buttons, text, and cards, ensuring a consistent, attractive look.

### 6. Database Interactions

- \*\*Collections\*\*:

- \*\*Users\*\*: Stores `\_id`, `name`, `email`, `password` (hashed), `role`, `status`, `points`, `location`.

- \*\*Barcodes\*\*: Stores `\_id`, `value`, `userId`, `timestamp`, `location`, `pointsAwarded`.

- \*\*Logic\*\*:

- Checks `Barcodes` for duplicates on scan.

- Updates `Users.points` on new scans.

- Admin actions modify `Users` and `Barcodes` accordingly.

---

## Technology Stack

- \*\*Frontend\*\*:

- React Native (v0.72.6)

- React Navigation (screen navigation)

- `react-native-vision-camera` (v4.6.4) with native `useCodeScanner` (barcode scanning)

- `@react-native-async-storage/async-storage` (persistent login)

- Axios (API requests)

- Expo (v49.0.21, Prebuild workflow, EAS Build, Push Notifications)

- \*\*UI/UX Libraries\*\*:

- `react-native-paper` (Material Design components, theming for light/dark modes)

- `@rneui/themed` (additional UI components)

- `react-native-vector-icons` (icons, including bulb for dark mode toggle)

- \*\*Backend\*\*:

- Node.js with Express (API server)

- MongoDB (database)

- Mongoose (MongoDB ORM)

- JWT (authentication)

- `bcryptjs` (password hashing)

- `cors` (cross-origin support)

- `expo-server-sdk` (push notifications)

- \*\*Database\*\*: MongoDB (storing users and barcodes)

---

## Setup Notes

- \*\*Prebuild Workflow\*\*: Native folders (`android`, `ios`) are managed by Expo Prebuild. Add `/android` and `/ios` to `.gitignore` and regenerate with `npx expo prebuild`. Use `eas build` for APK generation.

- \*\*EAS Configuration\*\* (`eas.json`):

- Specifies `"cli.appVersionSource": "remote"` for versioning from `app.json`.

- Defines `preview` profile for Android APK builds.

- \*\*App Configuration\*\* (`app.json`):

- Includes plugins for `react-native-vision-camera` (with `useCodeScanner`) and `expo-notifications`.

- Sets Android package and camera permission.

- Configures splash screen and owner details.

- \*\*Installation\*\*:

- Frontend: Initialize with Expo, install navigation, barcode scanning, UI, and notification libraries.

- Backend: Set up Express server with MongoDB, authentication, and notification dependencies.

---

## UI/UX Design Guidelines

- \*\*Attractive Design\*\*: Leverage `react-native-paper` for Material Design components (e.g., FABs, Cards) and `@rneui/themed` for modals and search bars.

- \*\*Theming\*\*:

- Light Mode: White background (`#ffffff`), dark text (`#000000`), vibrant accents (e.g., `#6200EE`).

- Dark Mode: Dark background (`#121212`), light text (`#ffffff`), muted accents (e.g., `#BB86FC`).

- Bulb icon toggles themes, ensuring high-contrast text visibility.

- \*\*Consistency\*\*: Apply a unified color scheme across all screens, adjusted for light/dark modes.

- \*\*Responsive Layout\*\*: Use Flexbox for adaptable layouts on various screen sizes.

- \*\*Icons\*\*: Utilize `react-native-vector-icons` (e.g., bulb for theme toggle, trash for delete) for intuitive navigation.

- \*\*Animations\*\*: Incorporate subtle transitions (e.g., fade-ins, slide-ups) for a smooth experience.

- \*\*User Feedback\*\*: Display loading spinners and success/error toasts for interactions.

---