LocalQR Web Application for Travelers

Develop a web application called \*\*LocalQR\*\* aimed at helping travelers create, organize, and share lists of location-based recommendations through unique, user-specific URLs and QR codes. The application will have two main user flows: a \*\*dashboard\*\* for registered users to create and manage content, and a \*\*public guest view\*\* for people accessing the shared lists via QR codes or direct links. Below are the comprehensive requirements.

---

### Project Overview

The application will allow users to:

1. \*\*Register, log in, and manage their profile\*\* with location-based recommendations.

2. \*\*Create lists\*\* for specific cities or locations (e.g., “Paris Guide”), organizing entries into categories like “Places to Visit” or “Food & Beverage.”

3. \*\*Generate unique, shareable URLs and QR codes\*\* for each profile and list (e.g., `www.localQR.earth/username` for profiles and `www.localQR.earth/username/listname` for individual lists).

4. \*\*Track analytics\*\* for QR code scans and link clicks.

5. \*\*Display an interactive, mobile-friendly view\*\* for guests scanning QR codes or accessing URLs.

---

### UI/UX Requirements Based on Figma Designs

#### 1. \*\*Login and Sign-Up Pages (Dashboard)\*\*

\*\*Database Tables Required\*\*:

- \*\*Users\*\* for authentication details (email, password).

- \*\*Login Page\*\*: Fields for email and password, login button, and links for "Forgot Password" and "Sign-Up."

- \*\*Sign-Up Page\*\*: Fields for email, password, confirm password, phone number, and account type selection.

- \*\*Password Reset Page (Optional)\*\*: Allows users to request a password reset link if needed.

#### 2. \*\*Dashboard Pages for Registered Users\*\*

- \*\*Profile Page\*\*:

- \*\*UI/UX Components\*\*: Allows users to update profile picture, name, bio, account type, primary location, and add social media links with visibility toggles.

- \*\*Database Tables Required\*\*:

- \*\*Accounts\*\* for storing profile details (`username`, `name`, `bio`, `account\_type`, `profile\_picture`, `location\_id`).

- \*\*Social\_Media\*\* for storing social links with visibility settings.

- \*\*Cities\*\* and \*\*Countries\*\* for location-based autofill data.

- \*\*Data Flow\*\*: User updates are saved to `Accounts`, `Social\_Media` for links, and location data is retrieved from `Cities` and `Countries`.

- \*\*Favorites Page\*\*:

- \*\*UI/UX Components\*\*: Displays the user's favorite places, organized by categories (e.g., Food & Beverage, Adventure). Allows users to add new places to lists with cover images.

- \*\*Database Tables Required\*\*:

- \*\*Favourites\*\* for linking `user\_id` to `place\_id` and associating each favorite with a specific `list\_id`.

- \*\*Places\*\* for storing details about each favorite place (`name`, `address`, `category\_id`, etc.).

- \*\*Categories\*\* for organizing places within favorites.

- \*\*Lists\*\* for creating and organizing lists of favorites.

- \*\*Data Flow\*\*: Favorites are retrieved from `Favourites` and organized by `Categories`. Lists are managed in `Lists` and linked to favorites.

- \*\*Analytics Page\*\*:

- \*\*UI/UX Components\*\*: Shows QR code and link performance metrics (scans, clicks, CTR), with top-clicked locations and time-based activity.

- \*\*Database Tables Required\*\*:

- \*\*Analytics\*\* for tracking `click\_count`, `scan\_count`, `CTR`, with date filtering.

- \*\*QRs\*\* for linking analytics data to specific profiles or lists.

- \*\*Lists\*\* and \*\*Places\*\* for associating analytics with specific content.

- \*\*Data Flow\*\*: The page pulls analytics data from `Analytics` and links it to specific QR codes in `QRs`, displaying performance per list or place.

- \*\*QR & Link Management Page\*\*:

- \*\*UI/UX Components\*\*: Displays all QR codes generated for the profile and individual lists, with options to download.

- \*\*Database Tables Required\*\*:

- \*\*QRs\*\* for storing QR codes for profiles (`type="profile"`) and individual lists (`type="list"`), with `link` and `image`.

- \*\*Lists\*\* for linking `list\_id` to each QR code.

- \*\*Data Flow\*\*: QR codes are generated, linked to URLs (e.g., `www.localQR.earth/username` or `www.localQR.earth/username/listname`), and stored in `QRs`.

---

#### 3. \*\*Guest View Pages (Accessible via QR or Link)\*\*

- \*\*Profile View (Public List Display)\*\*:

- \*\*UI/UX Components\*\*: Displays user’s profile (name, photo, bio) and their lists, organized by categories (e.g., “Places to Visit”).

- \*\*Database Tables Required\*\*:

- \*\*Accounts\*\* for user profile details.

- \*\*Lists\*\* for retrieving list details and organizing places.

- \*\*Places\*\* and \*\*Categories\*\* for displaying places in each list under categories.

- \*\*Data Flow\*\*: The profile view fetches `Accounts` based on `username` from the URL, loads lists from `Lists`, and organizes places by `Categories`.

- \*\*Detailed Entry View\*\*:

- \*\*UI/UX Components\*\*: Shows detailed information about a selected place, including address, timings, and contact options.

- \*\*Database Tables Required\*\*:

- \*\*Places\*\* for storing and displaying place-specific details.

- \*\*Social\_Media\*\* for contact information and links.

- \*\*Categories\*\* for organizing the type of place.

- \*\*Data Flow\*\*: Places are loaded by `place\_id`, with contact info from `Social\_Media` and category labels from `Categories`.

- \*\*Interactive Map View\*\*:

- \*\*UI/UX Components\*\*: Displays all listed places on a map with clickable pins for details.

- \*\*Database Tables Required\*\*:

- \*\*Places\*\* for `latitude` and `longitude` to place pins on the map.

- \*\*Cities\*\* and \*\*Countries\*\* for location-based filters.

- \*\*Data Flow\*\*: The map retrieves `Places` data for pin locations, with filtering by `Cities` and `Countries`.

---

### Database Structure (with Route Support)

#### Key Tables:

- \*\*Users\*\*: Manages user credentials.

- \*\*Accounts\*\*: Stores user profiles with `username` for unique profile URLs.

- \*\*Social\_Media\*\*: Links social links with visibility.

- \*\*Favourites\*\*: Connects users to favorite places.

- \*\*Categories\*\*: Organizes place types.

- \*\*Analytics\*\*: Tracks QR code engagement.

- \*\*Countries\*\* and \*\*Cities\*\*: Supports location autofill.

- \*\*QRs\*\*: Manages QR codes and their links.

- \*\*Lists\*\*: Stores lists with unique `slug` for list-specific URLs.

- \*\*Places\*\*: Holds place details.

- \*\*List\_Place\*\*: Junction for many-to-many between lists and places.

---

### Route and QR Code Management

1. \*\*Profile Route\*\*:

- Each user has a unique profile URL: `www.localQR.earth/username`.

- The `username` is stored in `Accounts.username` and must be unique.

2. \*\*List/City-Specific Route\*\*:

- Each list has a unique URL: `www.localQR.earth/username/listname`.

- The `slug` (URL-friendly list title) is stored in `Lists.slug`, ensuring unique URLs.

3. \*\*QR Code Generation\*\*:

- Each profile and list has a QR code linked to the URL, stored in `QRs.link`.

- The application redirects scans to the correct profile or list.

---

### Application Logic

1. \*\*Login/Sign-Up Flow\*\*:

- Uses `Users` for credentials and `Accounts` for profile setup.

2. \*\*Profile Management\*\*:

- Updates to profile details are saved in `Accounts`, with social media links in `Social\_Media`.

3. \*\*List and Favorites Management\*\*:

- Users create lists in `Lists`, add places from `Places`, and organize with `Categories`.

4. \*\*Analytics Tracking\*\*:

- Each scan or click updates `Analytics`, linked to `QRs` for performance tracking.

5. \*\*Guest View Access\*\*:

- Public views load based on `username` and `slug`, displaying profiles, lists, or places.

---

### Additional Requirements

1. \*\*Database Constraints\*\*:

- Unique constraints on `username` in `Accounts` and `slug` in `Lists` to ensure route uniqueness.

2. \*\*API Endpoints\*\*:

- Provide endpoints for managing profiles, lists, analytics, and QR generation.

3. \*\*Google Maps API\*\*:

- For location-based autofill and interactive map display.

4. \*\*Mobile Optimization\*\*:

- Ensure guest views are optimized for mobile.