**NEAR App: Lights by the Lake Event Trail Prototype - Replit AI Specification Document**

### Objective

To build a fully functional **web-based prototype** of the **Lights by the Lake event experience** for NParks using **Replit AI**, with the aim of migrating or integrating key components into the React Native NEAR App after validation.

The prototype must simulate user interaction with the event map, including scanning QR codes, taking photos, sharing content, and viewing community engagement.

### Map Feature Requirements

**Location Map**: Jurong Lake Gardens, including: - Chinese Garden - Japanese Garden - Science Park Trail

**Map Base Layer Reference**: Use OneMap API (Singapore’s official mapping service) to render the map. - Prefer Leaflet or Mapbox JS for ease of integration. - Predefine the bounds and zoom around Jurong Lake Gardens.

**Pins / Markers** to be shown on the map: 1. **QR Trail Points (Science Park Trail)** - 8–10 pins with pre-defined coordinates - Pins should be interactable - Each pin represents a location to be visited by user

1. **Event Features**
   * Lantern Displays
   * Facilities (e.g., toilets, rest points)
   * Food Vendors (Night Bazaar)
   * Custom Pins with NParks branding

**Design Requirements**: - Use the **NEAR App color scheme** (green primary #00703c, white, grey secondary tones) - Include NParks logos and optional photo frame overlays for posts

### User Flow: Science Park Trail Challenge

1. User opens homepage
2. Clicks on “Lights by the Lake” Event Card
3. Enters Event Map Page
4. Sees Trail Map with QR Pin markers
5. Clicks on Pin > Modal/Bottom Sheet opens
6. Modal prompts:
   * **Simulated QR Scan** (button or image input)
   * **Take/Upload Photo** (use HTML file upload)
7. Once both actions are completed, mark location as completed (change pin icon)
8. Store data in Firebase: user ID, pin ID, timestamp, photo URL
9. Display confirmation and optionally prompt for social sharing
10. Once all QR pins are completed, display “Prize Available” banner

### Community Wall

* Shows user-submitted photos with location and timestamp
* Each post can be shared to FB/IG (via mock button links)
* Filter or tag by location/pin
* Users can view and like other posts (optional)

### Night Bazaar Vendor Rating Feature

* Some pins are vendor stalls
* Modal shows:
  + 5-Star Rating input (once per user)
  + Option to post feedback and photo to Community Wall
* Firebase saves vendor ID, rating, user ID, timestamp
* Admin dashboard to view/delete inappropriate content or rating disputes

### Admin Tools

Accessible via /admin route (no need for login, placeholder OK): - View community wall posts (with delete buttons) - View ratings per vendor (with delete option) - Export CSV of data for reporting (e.g. via Firestore to CSV script)

### Analytics Tracking Requirements

Store and display basic analytics (can be mocked or from Firebase): - Number of page views - Taps on floating action button (FAB) - Taps on pins - Number of QR pins completed per user - Number of active users - Engagement per user (actions taken)

### Implementation Notes for Replit AI

* Use **React JS (create-react-app)**
* Use **Firebase** for backend: Firestore, Storage, Auth (anonymous is fine)
* Use **OneMap API** to embed map tiles
* Use **Leaflet JS** for map rendering and marker management
* Store static pin data in a JSON file for prototype
* Allow image upload and preview using <input type='file'>
* All UI should match NEAR App design as closely as possible

### Optional Enhancements

* Add special photo frame overlay (static PNG layered over uploaded photo)
* Add QR code generator and validator (simulate scan success)
* Add user leaderboard (optional, can be omitted)

### Deliverables from Replit AI

1. Functional web app link with deployed version
2. Source code with clear folder structure:
   * /components
   * /pages
   * /firebase
   * /assets
3. Firebase Firestore data structure (Pins, Users, Posts, Ratings)
4. Admin interface at /admin

Let me know when you’re ready to refine the components or generate the initial JSON of pin locations to begin prototyping the Science Park Trail pins.