

Requirement 9

- **Description:** The system shall prevent duplicate vehicle events using debounce or cooldown logic.
- **Acceptance Criteria:**
 - Rapid repeated detections within the cooldown window do not create multiple events
 - Only one ENTRY or EXIT event is recorded per vehicle passage
 - Cooldown duration is configurable

Requirement 10

- **Description:** The system shall define and enforce a JSON schema for all occupancy update payloads.
- **Acceptance Criteria:**
 - All incoming payloads follow a documented JSON structure
 - Requests missing required fields are rejected

Requirement 12

- **Description:** The system shall validate incoming requests using a Firebase Cloud Function before writing to Firebase.
- **Acceptance Criteria:**
 - All write operations pass through a Cloud Function
 - Invalid payloads are rejected before database writes
 - Only validated data is stored in Firebase

Requirement 13

- **Description:** The system shall reject unauthorized or malformed requests with clear error responses.
- **Acceptance Criteria:**
 - Malformed requests return an appropriate HTTP error status
 - Error responses include a clear message describing the issue

Requirement 14

- **Description:** The system shall store all parking and event data with server-side timestamps.
- **Acceptance Criteria:**
 - All event records include a server-generated timestamp
 - Client-provided timestamps are not trusted for authoritative records
 - Stored timestamps reflect the server time of processing

Requirement 21

- **Description:** The system shall enforce occupancy limits such that $0 \leq \text{occupancy} \leq \text{capacity}$.
- **Acceptance Criteria:**
 - Occupancy never exceeds the defined capacity
 - Occupancy never drops below zero
 - Invalid updates that violate limits are rejected or corrected

Requirement 22

- **Description:** The system shall retrieve real-time occupancy updates in the web application within 1–3 seconds.
- **Acceptance Criteria:**
 - Occupancy updates are reflected in the web UI within 3 seconds of change
 - Data is retrieved via the Firebase GET endpoint
 - Performance requirement is met under normal load conditions

Requirement 33

- **Description:** Host Web Server on Firebase
- **Acceptance Criteria:**
 - The web application is deployed to Firebase Hosting
 - Core pages are displayed correctly
 - Pages are functional

Requirement 34

- **Description:** The system shall include a pre-written suggested prompt that users can select to ask the chatbot.
- **Acceptance Criteria:**
 - A visible suggested prompt is displayed in the chatbot interface
 - User can insert the suggested prompt into the input field with one action (click/tap)
 - Inserted prompt is editable before sending
 - Prompt submits successfully and returns a response from the chatbot

Requirement 35

- **Description:** The system shall provide a Firebase-backed authenticated GET endpoint that allows the web application to retrieve real-time occupancy data.
- **Acceptance Criteria:**
 - Sending a valid authenticated GET request returns current occupancy data
 - Response includes occupancy count, capacity, and lastUpdated timestamp
 - Unauthorized requests are rejected

Requirement 36

- **Description:** The website shall fetch occupancy data from the Firebase GET endpoint instead of using mock data.
- **Acceptance Criteria:**
 - Website displays live data returned from the GET endpoint
 - No mock or hardcoded occupancy values are used in production