k-vibez - Database - Rules

```
1
rules_version = '2';
2
service cloud.firestore {
3
  match /databases/{database}/documents {
4


5
    // ===== ADMIN COLLECTION =====
6
    // Super secure - oly accessible by authenticated admins
7
    match /admins/{adminId} {
8
      allow read: if request.auth != null &&
9
        request.auth.uid == adminId;
10
      allow write: if request.auth != null &&
11
        (request.auth.uid == adminId ||
get(/databases/$(database)/documents/admins/$(request.auth.uid)).data.role ==
'super_admin');
12
    }
13
```

```
14
    // ===== PAYMENTS COLLECTION =====
15
    // Development rules - secure but functional
16
    match /payments/{paymentId} {
17
      // Allow ANYONE to create payments (for ticket purchases)
18
      allow create: if
19
        // Basic validation to prevent spam/abuse
20
        request.resource.data.amount is number &&
21
        request.resource.data.amount > 0 &&
22
        request.resource.data.amount <= 1000000 && // Max 1,000,000 KSH
23
        request.resource.data.transactionReference is string &&
24
        request.resource.data.transactionReference.size() >= 6 &&
25
        request.resource.data.transactionReference.size() <= 20 &&
26
        // Required fields must exist
27
```

```
        request.resource.data.fullName is string &&
28
        request.resource.data.email is string &&
29
        request.resource.data.phoneNumber is string &&
30
        request.resource.data.paymentMethod is string &&
31
        // Auto-set status to pending
32
        request.resource.data.status == 'pending' &&
33
        // Auto-set timestamp
34
        request.resource.data.createdAt == request.time &&
35
        // Prevent overwriting existing documents
36
        !exists(/databases/$(database)/documents/payments/$(paymentId));
37


38
    // Allow users to read THEIR OWN payments (using phone/email match)
39
    // OR allow public read with payment ID (for confirmation pages)
40
    allow read: if
41
```

```
      // Anyone can read if they know the document ID
42
      true;
43
      // For production: request.auth != null ||
44
      // request.query.get('paymentId') == paymentId;
45


46
    // ===== ADMIN ACCESS =====
47
    // Admins can read all payments
48
    allow read: if request.auth != null &&
49
      exists(/databases/$(database)/documents/admins/$(request.auth.uid));
50


51
    // Admins can update payments (for status changes: pending → confirmed)
52
    allow update: if request.auth != null &&
53
      exists(/databases/$(database)/documents/admins/$(request.auth.uid)) &&
54
      // Can only update specific fields
55
```

```
request.resource.data.diff(request.resource.data).affectedKeys().hasOnly(['status',
'confirmedBy', 'confirmedAt', 'notes']) &&
56
        // Status can only go from pending to confirmed/rejected
57
        (request.resource.data.status == 'confirmed' ||
58
         request.resource.data.status == 'rejected' ||
59
         request.resource.data.status == 'pending');
60


61
      // Admins can delete payments (only super admins)
62
      allow delete: if request.auth != null &&
63
        get(/databases/$(database)/documents/admins/$(request.auth.uid)).data.role ==
'super_admin';
64
    }
65


66
    // ===== TALENTS COLLECTION =====
67
    match /talents/{talentId} {
68
      // Public can read talents
```

```
69
      allow read: if true;
70


71
      // Only admins can write
72
      allow write: if request.auth != null &&
73
        exists(/databases/$(database)/documents/admins/$(request.auth.uid));
74
    }
75
    match /payments/{paymentId} {
76
  allow create: if true; // Allow anyone to create payments
77
  allow read: if true;   // For now - change later for security
78
}
80
    // ===== EVENTS COLLECTION =====
81
    match /events/{eventId} {
82
      // Public can read events
83
      allow read: if true;
```

```
84


85
      // Only admins can write
86
      allow write: if request.auth != null &&
87
        exists(/databases/$(database)/documents/admins/$(request.auth.uid));
88
    }
89
  }
90
}
```