# 📋 Random Box Event Manager Page – Requirements Definition

## 🧩 1. Purpose

We have built an administrator page for operating the random box event, so that you can \*\*monitor\*\* and \*\*control\*\* in real time the event participation status, BTC payment history, user logs, event settings, etc.

---

## 🧱 2. Key Functional Requirements

### ✅ \[1] Dashboard

| Features | Description |

| ---------------- | --------------------------- |

| Total number of participants | Today/cumulative number of participants |

| BTC Payment Status | Whether BTC was paid today, payment time, recipient ID |

| Recent Participation Log | Recent 10 Participation Records (Reward, Time, User) |

| First Box vs Regular Box Ratio | Visualization of Today's Engagement Type Statistics |

| Check today's event settings | Check probability table, restrictions, etc. |

---

### ✅ \[2] View participation records

| Features | Description |

| ----------- | ---------------------------------- |

| View user participation log | Search by userId or date |

| Detailed fields | User ID, participation date and time, box type, reward result, BTC or not |

| Filter | Reward Amount, BTC Recipient, Specific Date |

| Export function | CSV or Excel download |

---

### ✅ \[3] BTC Payment Status Management

| Features | Description |

| --------------- | --------------------------------------- |

| Check if BTC is paid today | Show current payment status |

| Manual Reset | Admin can manually reset BTC payout status (e.g. for testing purposes) |

| Check auto-initialization status | Display logs for CloudWatch auto-reset success |

---

### ✅ \[4] Manage reward probability table (optional)

| Features | Description |

| ------- | ------------------------ |

| Probability Table Visualization | Current probability table UI for each box type/condition |

| Change probability value | Save after adjusting probability (applies upon administrator approval) |

| Change History | Log of who changed what probability, when |

---

### ✅ \[5] Admin account/log management

| Features | Description |

| --------- | ------------------------------ |

| Admin Login | Email/Password or JWT based login |

| Task Log | Check the task history log by administrator (e.g., executing initialization, etc.) |

| Administrator Privileges | General Administrator / Super Administrator |

---

## 🖥️ 3. UI/UX Requirements

| Item | Description |

| --------- | ----------------------------------------- |

| Framework | React.js (Next.js or CRA) |

| Authentication method | JWT (AccessToken), Cookie or Authorization header |

| Admin-only route | Routing to URIs below /admin |

| Real-time | WebSockets or periodic polling if needed |

---

## ⚙️ 4. System/Technical Requirements

| Item | Description |

| ------ | ------------------------------------------------------ |

| API Integration | Existing Serverless API (`/event`, /reset-btc, DynamoDB, etc.) |

| DB linking | userEventLogTable, btcStatusTable (DynamoDB) |

| Authentication | Admin token issuance and verification API required (`/auth/login`) |

| Security | Admin page cannot be accessed without login, HTTPS required |

---

## 🧪 5. Test Scenario (Summary)

| Scenario | Expected Results |

| -------------------------- | ------------------------- |

| Admin login successful | Dashboard access possible after JWT issuance |

| Added participation log while BTC was not paid | Only 1 BTC recipient shown |

| BTC participants reappear after manual initialization | Normal payment processing |

| Filter user engagement logs | Filtered precisely by specific date, amount, user ID |

---

## 🔧 6. Additional Considerations

| Item | Description |

| --------- | ----------------------------------------- |

| Real-time log processing | CloudWatch log integration or DynamoDB Streams available |

| Multilingual support | Considering possibility of Korean/English in management page UI (i18n) |

| Future expansion | We may also consider adding functions such as user suspension and event pause/resume. |