CODING CLUB'S RAJINIKANTH

Hey folks,
This is Devaesh from AIDS first year. So, I've chosen scripting as my option, but to be really honest, I tried working on app development and failed at it—not very badly, but badly.

Anyways, coming back to cricket, I put together this project called **"Coding Club's Rajinikanth"** (don't ask why I named it that—I was probably running low on sleep). But hey, just like a superstar, this one's got style. It's basically a *superstar managing system* with a smooth tap-to-register kind of flow—with a bit of swag too.

This project is all about making things fast, fun, and just a little bit flashy. Whether you're registering participants or managing events, it's meant to feel effortless and cool—like a smooth one-tap experience.

So that's it for the intro, I'll be putting down my "scripting-cum-coding" done in the last 3 days in the following sections of this PDF, with a little help from AI (not too much, but just a little much ;p).

PROBLEM STATEMENT:

Managing large-scale events involves chaotic check-ins, manual tracking, and resource mismanagement. The goal is to build an Event Management Application that simplifies this process using **a unique way of check-in** for attendees.

Key requirements:

- Upload attendee data via CSV and assign a **unique ID**.
- Enable **secure check-in** and prevent duplicate registrations.
- Track **lunch/kit distribution** and stop repeat claims.
- Provide a **dashboard** with real-time stats for organizers.

Our goals include **creative ID generation**, **offline support**, **sync across desks**, and generating **event reports** from the collected data.

PROJECT OVERVIEW & THE UNIQUE SOLUTION:

Our project is built to solve one of the most common problems in large-scale events—handling check-ins efficiently across multiple registration desks simultaneously. Traditional systems often struggle with duplicate entries, long queues, or technical limitations, and that's exactly where EventHub steps in with two standout solutions.

1. NFC-Based Seamless Check-in (Primary Innovation)

In today's world, most smartphones come equipped with an NFC chip. We leverage this by enabling attendees to simply tap their phone on an NFC reader at the registration desk to check in—no scanning, no delays, just one smooth tap. This makes the process incredibly fast, contactless, and secure. It's one of our most innovative features, especially useful for high-traffic areas with multiple desks.

2. NFT-Style Image Check-in (Backup + Style Factor)

But what if the attendee's device doesn't support NFC? Or what if the NFC hardware at the desk fails? We've got that covered with our second unique approach:
Each attendee receives a personalized, NFT-like check-in image. These stylized images (e.g., monkey-themed tokens) are visually unique and serve as an alternative check-in method. The organizer simply scans the NFT image, and the system registers the attendee.

This not only ensures redundancy but also adds a tech-savvy, modern twist to traditional QR codes. It makes the check-in experience fun, secure, and visually distinctive.

🔄 Additional Backup: Manual Entry

In case of full tech failure (no internet, broken devices, etc.), attendees can also check in manually by telling their unique Attendee ID at the desk. This ID is assigned during registration and ensures that every attendee always has a way to check in—regardless of tech availability.

KEY SOLUTIONS:

Our Project provides two core modules—Organizer Login and Attendee Login—ensuring smooth event execution with innovative solutions.

Organizer Login: Empowering Event Management: The Organizer Login provides a comprehensive suite of tools to manage every aspect of the event efficiently.

- Organizer Dashboard
  The dashboard offers a centralized view of key event metrics:
  1. Total Attendees, Registrations, Lunches Served, and Swags Distributed.
  2. Attendance Statistics: Tracks check-in rates with a progress bar. Visualizes lunch and swag distribution.
  3. Feedback Analysis: Displays rating distribution and recent feedback for actionable insights.
  4. Registered Attendees: A table lists attendee credentials, including Attendee ID, User ID, Name, Email, Phone, Role, QR Code Path, NFT Token ID, Check-In Status, and Lunch/Swag Collection Status.

- Add Attendee
  Organizers can manually add attendees by entering details such as Name, Email, Phone, and Role (e.g., Attendee or Speaker). Upon submission, attendees receive an automated email with their login credentials (username and password).
- Upload CSV
  For bulk registration, organizers can upload a CSV file with columns for Name, Email, Phone, and Role. This feature streamlines the process and triggers automated credential emails to attendees.
- Check-In, Lunch, and Swags (NFC/QR/NFT Integration)
  These features offer three innovative methods to manage check-ins and resource distribution:
    1. NFC Tapping: Attendees tap an NFC-enabled device to check in or collect lunch/swags.
    2. QR Scanning: If NFC fails, organizers can scan a QR code using the camera.
    3. Manual Entry: Enter the Attendee ID manually as a fallback option.
       Each method updates the dashboard in real-time, ensuring accurate tracking.

Attendee Login: Seamless Event Participation: The Attendee Login empowers participants with a personalized event experience:

- NFC Option: Attendees can use NFC tapping to check in and collect lunch/swags, ensuring a quick and hassle-free process.
- Unique NFT Token: Each attendee receives a unique NFT token, serving as a digital collectible and proof of participation that is to be scanned.
- Feedback Section: Attendees can provide feedback by rating the event (1-5 stars) and adding comments, which are reflected in the organizer's dashboard for analysis.
- Event Details: Attendees can view their NFT token, and status for check-in, lunch, and swag collection.

For a complete look at the UI and application flow, check out the image folder here:

(Web Application Screenshots)

TECHNICAL ARCHITECTURE AND TECH STACK:

The web app's architecture has three main parts that work together:

1. User Interface (Frontend):
    - What: Web pages for organizers (to manage attendees) and attendees (to see event info).

    - Pages:
        - Organizers: Login, dashboard (shows stats), add attendees, upload CSV, check-in, track lunch/swags.
        - Attendees: Login, view events/sessions, see NFT image, give feedback.
    - How: Users interact via browser, sending requests to the server.
2. Server (Backend):
    - What: The brain of EventHub, handling all logic and data.
    - Jobs:
        - Manages logins (secure with passwords).
        - Creates NFT images for check-ins (replaces QR codes).
        - Tracks check-ins, lunch, and swags.
        - Sends emails to attendees with login details.
        - Shows stats (e.g., how many checked in) and stores feedback.
    - How: Processes requests, talks to the database, and generates NFT images.
3. Database & Storage:
    - What: Stores all event data.
    - Data:
        - Organizers (login info).
        - Attendees (name, email, NFT image, check-in status).
        - Events & sessions (e.g., TechFest 2025 schedule).
        - Feedback (attendee ratings/comments).
    - Storage: Saves NFT images in a folder (public/nft_images).
    - How: Temporary database (resets on restart) stores data; images saved on server.

Tech Stack (Tools Used)

- Backend:
    - Node.js: Runs the server.
    - Express.js: Handles web requests (like clicks on pages).
    - SQLite: Stores data (like attendee info).
    - Canvas: Makes unique NFT images (monkey face, hat, colors).
    - Nodemailer: Sends emails to attendees.
    - Multer & csv-parser: Uploads and reads CSV files.
    - bcrypt: Secures organizer passwords.
    - express-session: Keeps users logged in.
- Frontend:
    - EJS: Builds web pages dynamically.
    - Tailwind CSS: Makes pages look nice and work on phones.
- Other:
    - npm: Installs tools.
    - Gmail: Sends emails (needs real account setup).
    - Filesystem: Stores NFT images.

FUTURE ENHANCEMENTS AND IMPROVEMENTS:

Our Project is already a powerful event management tool, but we have big plans to make it even better! By improving security, user experience, and technology, we aim to create a seamless and exciting platform for organizers and attendees at events like INVENTE.

Planned Improvements

1. Better NFT Tokens:
    - Current Issue: The NFT images (monkey-like with random features) are basic and not highly secure for registration.
    - Fix: Use advanced NFT generation with unique cryptographic IDs (e.g., blockchain-based tokens) to ensure top-notch security. Add more creative designs (e.g., glasses, fur styles) for a fun, personalized look.
2. Improved User Interface (UI/UX):
    - Current Issue: The UI works but could be more modern and user-friendly.
    - Fix: Redesign the interface with a sleek, intuitive layout using React or Vue.js. Add features like drag-and-drop CSV uploads, interactive dashboards, and mobile-friendly views to make navigation effortless and enjoyable.
3. Advanced NFC Integration:
    - Current Issue: The NFC toggle exists but needs a physical NFC reader for check-ins.
    - Fix: Build a custom NFC reader and code it to instantly recognize the NFC chips. This will enable fast, seamless check-ins without manual ID entry, using hardware like Raspberry Pi with NFC modules.

Additional Enhancements

4. Persistent Database:
    - Switch from in-memory SQLite to PostgreSQL for permanent data storage, ensuring no data loss on server restarts.
5. Feedback Analytics:
    - Add a dashboard to visualize attendee feedback (e.g., average session ratings, word clouds of comments) for better event planning.
6. Mobile App:
    - Develop a mobile app (iOS/Android) for attendees to view events, display NFTs, and submit feedback on the go.

Goal: Make it more secure, user-friendly, and tech-forward, ensuring unforgettable event experiences!

CONCLUSION:

So yeah, the project I've built so far ticks off most of what the problem statement asked for—and honestly, I'm kind of surprised at how well the model prototype turned out (not perfect but okish). Sure, it has its flaws, bugs, and "what-even-is-this" moments. I've attached my **GitHub repo** with the full code and this **script PDF**, so if you've made it this far, I genuinely hope you take a look at the project too. Three sleepless nights, and a lot of debugging despair into it—so please, treat it with kindness.

All that said, I really believe there's so much more I can do with this. Given time, I'm confident I can kill the bugs, polish the features, and maybe even turn this into something event-worthy managing system. Because scripts can be improved, bugs can be fixed, and even a failed attempt at app development can lead to something innovative and functional.

Thanks for scrolling till this last page. Catch you on the next tap-to-register revolution.