# EEVO

**Event & Election Virtual Organizer (Mobile App)**

**Tagline**: One platform for department events, cross-department sports & culturals, elections, finances, and notifications — following real college rules.



**Mentor:**

**TEAM MEMBERS**

## 📱 Abstract

EEVO Mobile is a cross-platform Flutter app that complements the web platform, offering students and staff a responsive, role-based interface to manage events, elections, and finances. It mirrors the full functionality of the web app while optimizing for mobile-first interactions, push notifications, and offline access.  
Now with **collaborative event creation**, EEVO enables departments and individuals to co-host events with shared visibility and authority.

## 📘 Project Description

EEVO Mobile is a cross-platform Flutter app designed to manage college events, elections, and finances. It complements the EEVO web platform by offering a mobile-first experience with responsive design and secure role-based access.  
The app supports departmental workshops, inter-departmental events, and student elections, all modeled on real college protocols to ensure compliance and transparency.  
Users are assigned roles—HOD, PT Sir, Event Coordinator, Election Commissioner, Treasurer, or Student—with tailored permissions. This enables decentralized management with centralized oversight.

### Core Features:

* **Event Management**: Create, approve, and monitor events, including collaborative events across departments or individuals.
* **Election Workflow**: Nomination to inauguration with secure voting and result analysis.
* **Finance Tracking**: Upload and verify bills linked to events/elections.
* **Push Notifications**: Real-time updates for roles, events, and elections.
* **Offline Access**: Local caching for low-connectivity environments.

The backend uses Java + Spring Boot, with secure REST APIs and JWT authentication. Mobile-specific endpoints ensure performance. CI/CD pipelines automate testing and deployment, and the app is distributed via Firebase App Distribution, Play Store, and TestFlight.

EEVO Mobile mirrors real-world college operations, supporting student governance, staff coordination, and institutional transparency.

## Tech Stack Summary

| **Layer** | **Technology** | **Purpose** |
| --- | --- | --- |
| Mobile Frontend | Flutter (Dart) | Cross-platform UI, client-side logic |
| Backend | Java + Spring Boot | REST APIs, business logic |
| Database | MySQL / PostgreSQL | Structured data storage |
| ORM | Spring Data JPA / Hibernate | Object-relational mapping |
| API Docs | Swagger / SpringDoc OpenAPI | Auto-generated API documentation |
| Authentication | Spring Security + JWT | Secure login, role-based access |
| CI/CD | GitHub Actions + Codemagic | Automated testing and deployment |
| Deployment | Firebase / Play Store / TestFlight | Mobile app hosting and distribution |

## Key Roles & Authorities

| **Role** | **Responsibilities** |
| --- | --- |
| HOD | Schedule elections, preload staff IDs, manage staff registry, approve configurations |
| Department President | Create/modify departmental events, authorize organizers, approve collaborative events |
| Association Member | Create/modify events when permitted |
| Election Commissioner | Manage election lifecycle, assign post-election roles |
| PT Sir | Create/manage sports events, assign Sports Leaders |
| Sports Leader | Manage sport-specific activities |
| Event Coordinator | Manage cultural events (appointed by Principal) |
| Treasurer | Upload bills/receipts, manage finance |
| Staff/Student Viewer | View-only access; voting rights as per rules |
| Event Organizer (Staff) | Create/mentor career guidance/workshops |

## 🔄

## Collaborative Event Feature

### 📌 Feature Overview

EEVO supports **inter-departmental and individual collaborations** during event creation. Organizers can select one or more departments or individuals to co-host an event. This ensures shared visibility, joint responsibility, and synchronized authority across all collaborators.

### 🧩 Functional Highlights

* **Collaboration Toggle**: Enable “Collaborate with…” during event creation.
* **Dual Visibility**: Event appears in dashboards of all selected departments and their Presidents.
* **Shared Authority**:
  + Both Presidents can approve, modify, or cancel the event.
  + Finance uploads and notifications are visible to all collaborators.
* **Audit Trail**: Collaboration metadata is logged for transparency and future audits.

### Role-Based Impact

| **Role** | **Impact** |
| --- | --- |
| Department President | Can initiate or approve collaborative events |
| Association Member | Can propose collaboration if permitted |
| Treasurer | Sees shared finance entries across departments |
| Event Organizer | Manages cross-departmental logistics |
| Staff/Student Viewer | Sees events relevant to their department or collaborations |

## Requirements Analysis

### Functional Requirements:

* Role-based event creation and management
* Election eligibility and scheduling
* Push notifications for events and election stages
* Finance uploads and payment tracking
* Year advancement and audit logging
* **Collaborative event creation with shared visibility and authority**

### Non-Functional Requirements:

* Offline-first architecture
* Secure authentication and role-based authorization
* Responsive UI for Android and iOS
* CI/CD integration for mobile builds
* API integration with backend services

## 🧪 Election Workflow

1. **Nomination**: 3-day window for 4th-year students
2. **Campaigning**: 5 days with enforced protocols
3. **Analysis**: 1 day for validation
4. **Voting**: 24-hour secure ballot
5. **Result**: Auto-analysis + manual role assignment
6. **Inauguration**: Scheduled by Commissioner
7. **Year Rollover**: Admin-confirmed or auto-incremented

## Design Highlights

* Role-based navigation with dynamic menus
* Secure voting interface with real-time status
* Push notification integration
* Offline caching for event and election data
* Finance upload with image and PDF support
* Configurable election rules and timers
* **Collaborative event creation UI with multi-select department/individual picker**
* **Event cards display collaboration badge and shared approval status**

## Detailed Implementation

### 🔹 Mobile Frontend

* **Tech**: Flutter (Dart)
* **Structure**:
  + main.dart: App entry point with routing and theme
  + screens/: Role-based screens (HOD, PT Sir, Commissioner, etc.)
  + widgets/: Reusable UI components (cards, forms, buttons)
  + models/: Data classes for events, elections, users
  + services/: API integration and local storage
* **Routing**: Navigator with named routes and guarded access
* **State Management**: Provider or Riverpod for reactive UI
* **Validation**: Form validation with role-based logic
* **Offline Support**: SQLite or Hive for local caching
* **Push Notifications**: Firebase Cloud Messaging (FCM)
* **Collaborative Events**:
  + Multi-select dropdown for departments/individuals
  + Collaboration badge in event cards
  + Shared approval logic in UI

### API Docs

* **Tech**: Swagger / SpringDoc OpenAPI
* **Mobile Integration**: Auto-generated client using swagger-codegen or manual HTTP calls via http or dio package

### 🔹 Backend

* **Tech**: Java + Spring Boot
* **Mobile-specific endpoints**:
  + /api/mobile/events
  + /api/mobile/elections
  + /api/mobile/notifications
* **Authentication**: JWT-based auth with mobile token refresh
* **Collaborative Events**:
  + Extend event schema with collaborators[]
  + Normalize with junction table: event\_collaborators(event\_id, dept\_id, user\_id)
  + Update fetch logic to include all relevant departments

### Authentication

* **Tech**: Spring Security + JWT
* **Flow**:
  + Login → Token → Secure API calls
  + Role-based UI rendering
  + Token refresh and logout handling

### 🔹 CI/CD

* **Tech**: GitHub Actions + Codemagic (optional)
* **Pipeline**:
  + Build APK and IPA on push
  + Run unit and widget tests
  + Deploy to Firebase App Distribution or TestFlight
  + Secrets managed via GitHub repository settings

### 🔹Deployment

| **Layer** | **Platform** |
| --- | --- |
| Backend | Render / Railway |
| Mobile App | Firebase App Distribution / Play Store / TestFlight |
| Database | Railway / Supabase (optional) |

## 🎯Outcome

* Real-world compliance
* Centralized control for sports/cultural events
* Transparent finance tracking
* Auditable election results
* Mobile-first access for students and staff
* **Collaborative event management across departments and individuals**

## 🚀

## Future Scope

* Biometric login (fingerprint/face)
* Offline voting with sync-on-connect
* Role-based analytics dashboard
* PWA fallback for browser access
* **Cross-campus collaboration for inter-college events**
* **Dynamic role delegation for collaborative events**